K + K = 120

Papers dedicated to László Kálmán and András Kornai
on the occasion of their 60th birthdays

K + K = 120

Papers dedicated to László Kálmán and András Kornai
on the occasion of their 60th birthdays

Revised and extended edition

edited by

Beáta Gyuris
Katalin Mády
Gábor Recski

MTA Research Institute for Linguistics
Budapest, 2019

Revised and extended edition of the 2017 online version.

Available online at www.nytud.hu/kk120

ISBN 978-963-9074-82-8 (paperback)
ISBN 978-963-9074-83-5 (pdf)
DOI: 10.18135/kk120.2019

Published by MTA Research Institute for Linguistics
P.O. Box 360, H–1394 Budapest, Hungary
www.nytud.hu

Cover design by András Rung.
Cover image: András Rung, *Stanford landscape*, 2019, © András Rung.
Typeset by Zoltán G. Kiss, system LaTeX, main text in 11/13 CMU Serif, titles in Barlow.

Printed in Hungary by Efo Kiadó és Nyomda Kft., Százhalombatta.

Contents

Tabula gratulatoria

*Ács, Judit * Alexin, Zoltán * Anthony, Jan * Asudeh, Ash * Axelrod, Amittai * Bakró-Nagy, Marianne * Baló, András Márton * Bánréti, Zoltán * Bárkányi, Zsuzsanna * Beaver, David I. * Beke, Tibor * Bende-Farkas, Ágnes * Bernstein, Jared * Bimbó, Katalin * Biró, Tamás * Blaho, Sylvia * Borbély, Gábor * Both, Zsolt * Bródy, Mihály * Bukatin, Michael * Cser, András * Dekker, Paul * Dékány, Éva * Deme, Andrea * Domokos, Johanna * Dyekiss, Emil Gergely * É. Kiss, Katalin * Fejes, László * Fenyvesi, Anna * Forró, Orsolya * Fulop, Sean * Füle, Ágnes * G. Kiss, Zoltán * Gärtner, Hans-Martin * Georgieva, Ekaterina * Gibson, Maik * Gyenis, Zalán * Gyuris, Beáta * Hetényi, Eszter * Hollósi, József * Jäger, Gerhard * Kádár, Edit * Kaisse, Ellen * Káldi, Tamás * Kas, Bence * Kenesei, István * Kertész, János * Kiefer, Ferenc * Kiss, Angelika * Koskenniemi, Kimmo * Kowalski, Tomasz * Kracht, Marcus * Kuutti, Pirkko * Laczkó, Tibor * van Leusen, Noor * Lipp, Veronika * Lukács, András * Lukács, Ágnes * Madarász, Levente * Mády, Katalin * Magyar, Lilla * Makrai, Márton * Maleczki, Márta * Martín-Vide, Carlos * Mészáros, Éva * Mittelholcz, Iván * Molnár, Cecília Sarolta * Molnár Zoltán * Mus, Nikolett * Muskens, Reinhard * Nádasdy, Ádám * Nagy, Viktor * Nemeskey, Dávid * Németh T., Enikő * Oravecz, Csaba * Orsós, Anna * Pajkossy, Katalin * Palmer, Martha * Patay, Fanni * Pléh, Csaba * Polgárdi, Krisztina * Prószéky, Gábor * Pullum, Geoffrey K. * Ransil, Pat * Rebrus, Péter * Recski, Gábor * Rehm, Georg * Reichel, Uwe D. * Rónyai Lajos * Rung, András * Sakarovitch, Jacques * Schepácz, András * Shieber, Stuart * Simon, Eszter * Siptár, Péter * Soria, Claudia * Surányi, Balázs * Szabó, Ildikó Emese * Szabó, Zoltán Gendler * Szabolcsi, Anna * Szakadát, István * Szécsényi Krisztina * Szécsényi Tibor * Szigetvári, Péter * Thompson, Paul * Tilesch, Gábor * Törkenczy, Miklós * Turner, Ken * Yli-Jyrä, Anssi * Vardi, Moshe * Varga, László * Vámos, Tibor * Vásárhelyi, Dániel * Veltman, Frank * Wintner, Shuly * Zuber, Richard * Zvolenszky, Zsófia * Zséder, Attila*

Foreword

László Kálmán and András Kornai, two outstanding linguists who have influenced generations of students and colleagues around the world, celebrated their 60th birthdays in 2017. This collection is a revised and extended version of the online Festschrift *K+K=120* (www.nytud.hu/kk120), published for the occasion in 2017, featuring papers written by their students, friends, collaborators, and other scholars who, by being part of the project, wanted to express their appreciation for András and László.

László Kálmán's main interests lie in the theory of grammar, formal semantics and computational linguistics, but there is virtually no subdomain of linguistics that he hasn't tried his hands at. He has written about phonology, morphology, (lexical) syntax, pragmatics, as well as problems of compositionality, analogy-based learning, monotonicity, and commonsense reasoning. He coauthored and edited a volume on descriptive Hungarian syntax (2001), coauthored a descriptive grammar of Boyash (2009, with Anna Orsós) and several textbooks (in 2002 with Gábor Rádai, and in 2007 with Viktor Trón, among others). He has also made significant contributions to the popularization of science and to reforming the teaching of Hungarian grammar at Hungarian primary and secondary schools.

András Kornai has, since the early 1980s, pursued research at the intersection of mathematics and linguistics. Over the past four decades he made influential contributions to all major areas of (computational) linguistics, some notable examples of which are his formal construction of X-bar theory with Geoffrey K. Pullum (1990), and his monographs on morphology (1994), phonology (1995), mathematical linguistics (2007), and semantics (2019). Over the last 15 years, András has also established a new school of human language technologies in Budapest, mentoring and leading an interdisciplinary team of mathematicians, linguists and engineers.

András and László, beyond their extensively overlapping interests across a broad range of research topics, also share numerous rare scholarly and human qualities. These include their outstanding intellectual abilities, which enabled them to become experts in an exceptionally wide range of domains spanning linguistics, mathematics and computer science. Further, both of them are known for their rigorous standards and uncompromising critical attitude, which, however, they are ready to apply to themselves as much as to others. All this already showcases László and András as creative scholars; but their spark and flair bear mentioning also: colleagues and students of theirs have always appreciated their witty, open-minded, approachable and entertaining personalities, and their ability to pursue interesting conversations with anyone on just about any topic. These qualities, in addition to their intellectual abilities and enthusiasm for teaching, have made them inspiring teachers and mentors for many generations of students, some of whom appear as authors in this collection. The diversity of topics appearing in this Festschrift is illustration of the variety of fields on which the work of Kálmán and Kornai has made an impact, including virtually all branches of theoretical linguistics and their intersections with mathematics and computer science.

The number and scope of manuscripts received was vast and the editors are greatly indebted to all those who have supported the seemingly endless process of compiling this collection. For reviewing the contributions to the volume, we thank all the authors as well as Thomas Graf, Péter Rebrus, Paul Thompson, Dániel Varga, Dániel Vásárhelyi, Richard Zach, and Zsófia Zvolenszky. We are also grateful to Zoltán G. Kiss for typesetting the volume, to Uwe Reichel for technical assistance, to Zsófia Zvolenszky for her help with the current Foreword, and to András Rung for the painting on the cover (prepared for this occasion) and for the cover design. For their help with the organization of the birthday workshop, for financial support for the production of the Festschrift and for taking on the role of the publisher we wish to express our gratitude to the MTA Research Institute for Linguistics, particularly to Veronika Lipp, Gábor Prószéky, Ágnes Talián, Judit Temesvári and Tamás Váradi.

Budapest, May 2019

the editors

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 1–18

Building word embeddings from dictionary definitions

JUDIT ÁCS
Budapest University of Technology and Economics;
Institute for Computer Science and Control,
Hungarian Academy of Sciences
judit@aut.bme.hu

DÁVID NEMESKEY
Institute for Computer Science and Control,
Hungarian Academy of Sciences
nemeskey.david@sztaki.mta.hu

GÁBOR RECSKI
Budapest University of Technology and Economics
recski@aut.bme.hu

KEYWORDS

ouroboros
natural language processing
semantics
lexicon
word embedding

ABSTRACT

We present an experimental method for mapping English words to real-valued vectors using entries of a large crowd-sourced dictionary. Intuition tells us that most of the information content of the average utterance is encoded by word meaning (Kornai 2010 posits 85%), and mappings of words to vectors (commonly known as *word embeddings*) have become a core component of virtually all natural language processing (NLP) applications over the last few years. Embeddings are commonly constructed on the basis of large corpora, approximating the semantics of each word based on its distribution. In a set of pilot experiments we hope to demonstrate that dictionaries, the most traditional genre of representing lexical semantics, remain an invaluable resource for constructing formal representations of word meaning.

1. Background

Nearly all common tasks in natural language processing (NLP) today are performed using deep learning methods, and most of these use *word embeddings* – mappings of the vocabulary of some language to real-valued vectors of fixed dimension – as the lowest layer of a neural network. While many embeddings are trained for specific tasks, the generic ones we are interested in are usually constructed with the objective that words with similar distributions (as observed in large corpora) are mapped to similar vectors. In line with the predictions of the *distributional hypothesis*, this approach causes synonyms and related words to cluster together. As a result, these general-purpose embeddings serve as robust representations of meaning for many NLP tasks; however, their potential is necessarily limited by the availability of data. Lack of training data is a major issue for all but the biggest languages, and not even the largest corpora are sufficient to learn meaningful vectors for infrequent words. Lexical resources created manually, such as monolingual dictionaries, may be expensive to create, but crowdsourcing efforts such as Wiktionary or UrbanDictionary provide large and robust sources of dictionary definitions for large vocabularies and – in the case of Wiktionary – for many languages.

Recent efforts to exploit dictionary entries for computational semantics include a semantic parser that builds concept graphs from dependency parses of dictionary definitions (Recski 2016; 2018) and a recurrent neural network (RNN) architecture for mapping definitions and encyclopaedia entries to vectors using pre-trained embeddings as objectives (Hill et al. 2016). In this paper we construct embeddings from dictionary definitions by encoding directly the set of words used in some definition as the representation of the given headword. We have shown previously (Kornai et al. 2015) that applying such a process iteratively can drastically reduce the set of words necessary to define all others. The extent of this reduction depends on the – possibly non-deterministic – method for choosing the set of representational primitives (the defining vocabulary). The algorithm used in the current experiment will be described in section 2. Embeddings are evaluated in section 3, section 4 presents our conclusions.

2. Word vectors from dictionary definitions

In this research, we eschew a fully distributional model of semantics in favor of embeddings built from lexical resources. At first glance, the two approaches seem very different: huge corpora and unsupervised learning vs. a hand-crafted dictionary of a few hundred thousand entries at most. Looking closer, however, similarities start to appear. As mentioned previously, generic ("semantic") embeddings are trained in such a way that synonyms and similar words cluster together; not unlike how definitions paraphrase the definiendum into a synonymous phrase (Quine 1951). The two methods thus can be viewed as two sides of the same empirical coin; we might not fully go against Quine then when we "appeal to the nearest dictionary, and accept the lexicographer's formulation as law". Representing (lexical) semantics as the connections between lexical items has a long tradition in the NLP/AI literature, including Quillian's classic Semantic Memory Model (Quillian 1968), widely used lexical ontologies such as WordNet (Miller 1995) and recent graph-based models of semantics such as Abstract Meaning Representations (Banarescu et al. 2013) and 4lang (Kornai 2012; Recski 2018).

In the model presented below, word vectors are defined not by count distributions (as in e.g., Pennington et al. 2014), but by interconnections between words in the dictionary. For the purpose of this paper, we chose the English Wiktionary[1] as the basis of our embedding, because it is freely available; however, the method would work on any monolingual dictionary. The word vectors are computed in three steps.

First, we preprocess the dictionary and convert it into a formal structure, the *definition graph*: a directed graph whose vertices correspond to headwords in the dictionary. Two vertices A and B are connected by an edge $A \leftarrow B$ if the definition of the head contains the tail, e.g., *A: B C D*. Definition graphs can be *weighted* and *unweighted*. In the former, each vertex distributes the unit weight among its in-edges equally; in the latter, each edge has a weight of 1. Continuing the previous example, the edges from B, C and D to A have a weight of $\frac{1}{3}$ in the weighted graph and 1 in the unweighted one.

Next, an iterative algorithm is employed to find an "Ouroboros" set of words, which satisfies two conditions:

1. the whole vocabulary can be defined in terms of it, i.e., all directed paths leading to a word in the definition graph can be traced back to the Ouroboros set

2. it can define itself, so no words outside the set appear in the definitions of its members (we call this self-containedness the *ouroboros property*).

The idea that a small set of primitives could be used to define all words in the vocabulary is not new (Kornai 2018); several such lists exist. The most relevant to the current work is probably the Longman Defining Vocabulary (LDV), used exclusively in the definitions of earlier versions of the Longman Dictionary of Contemporary English (LDOCE) (Bullon 2003). The LDV is not minimal, and in previous work it served as our starting point to reduce the size of the essential word set as much as possible (Kornai

[1] https://www.wiktionary.org/

et al. 2015). Here we chose a different approach, not least because no such list exists for Wiktionary.

Finding the Ouroboros set would be easy if the definition graph was a DAG. However, due to the interdependence of definitions in the dictionary, the graph contains (usually many) cycles. Our algorithm deals with this by choosing a “defining” node in each cycle, and collecting these in a set. Then, all arrows from outside of the set to inside it are removed. It is clear that this set is defining, as every non-member vertex is reachable from the nodes in it. Furthermore, after the removal of inbound edges, the set satisfies the second condition and therefore it is an Ouroboros.

Trivially, the whole dictionary itself is an Ouroboros set, provided that dangling edges (corresponding to words in definitions that are themselves not defined in the dictionary) are removed from the definition graph.[2] Needless to say, we are interested in finding the smallest possible (or at least, a small enough) set that satisfies the property.

Mathematically inclined readers might recognize our Ouroboros as the *feedback vertex set* of the definition graph. In the remainder of this paper, we shall stick to the former (perhaps inaccurate) name, as it also hints at the way it is generated – see section 2.2. Furthermore, elements of the set shall be referred to – perhaps even more incorrectly than the singular term – with the plural form, *ouroboroi*.

In the final step of the algorithm, the vertices of the definition graph are mapped into real valued vectors in $\mathbb{R}^n$, where n is the size of the Ouroboros set. The vectors that correspond to the ouroboroi serve as the basis of the vector space; they are computed from the structure of the Ouroboros subgraph. Other words are assigned coordinates in this space based on how they are connected to the ouroboroi in the definition graph.

It is worth noting that in our case, the dimensionality of the embedding is dictated by the data; this is in sharp contrast to regular embeddings, where n is a hyperparameter.

The steps are explained in more detail below.

2.1. Preprocessing the dictionary

A dictionary is meant for human consumption, and as such, machine readability is, more often than not, an afterthought. Wiktionary is no exception, although its use of templates makes parsing a bit easier. We used the

[2] This move might sound dubious, but justifiable if the dictionary encompasses a large enough portion of the vocabulary of the language.

English dump of May 2017, and extracted all monolingual entries with the *wiktionary_parser* tool from the 4lang library.[3] The definitions are then tokenized, lemmatized and tagged for POS by the corresponding modules of the Stanford CoreNLP package (Manning et al. 2014).

At a very basic level, tokenization is enough to produce a machine readable dictionary. However, further transformations were applied to the dictionary to improve recall and decrease its size by removing irrelevant data, as well as to correct inconsistencies in how it was compiled. Raw word forms generally give low *recall* because of the difference in inflection between definienda and definientia. To solve this problem, we employed two essential techniques from information retrieval (IR): *lemmatization* and *lowercasing*. Our aim with the dictionary is to build a definition graph of common words. Looking at the dictionary from this angle, it is clear that it contains a large amount of irrelevant data.

- *Multiwords*: Wiktionary has entries for multiword units, such as expressions and noun compounds. While this poses no problems for the algorithm described below, currently we have no means to evaluate such lexical units.
- *Proper nouns*: proper nouns often cluster into strongly connected groups, such as mythologies (*Étaín*, *Midir* and *the Dagda*, amongst others, represent Ireland) or country-capital pairs (e.g., *Dehradun* and *Uttarakhand* from French India). Each such group inevitably “delegates” one of its members to the ouroboros, increasing its size for negligible gains.
- *Punctuation*: punctuation marks clearly have no role on the semantic level; on the syntactic side, our BOW approach renders them superfluous.
- *Stopwords*: similarly to punctuation, function words bring very little to the table; removing them is a common practice in IR.

We created a dictionary file for every combination of the transformations described above. Proper nouns and punctuation were filtered by their POS tags; stopwords according to the list in NLTK (Bird et al. 2009). In case of the latter two, not only were the tokens removed from the definitions, but their entries were also dropped. Table 1 lists the most important versions, as well as the effect of the various filtering methods on the size of the

[3] https://github.com/kornai/4lang/

vocabulary and the Ouroboros of the resulting dictionary. It can be seen that lowercasing and lemmatization indeed increase the recall, and that multiwords and proper nouns make up about one third of the dictionary. The effect on the size of the Ouroboros seems more incidental; it is certainly not linear in the change in vocabulary size.

Table 1: Effect of filtering steps on vocabulary and Ouroboros size

Preprocessing steps	Vocabulary size	Ouroboros size
none	175,648	3,263
lowercasing	176,814	3,591
lemmatization	179,212	3,703
no multiword	140,058	3,231
no proper nouns	151,652	2,688
no punctuation	175,651	3,263
no stopwords	171,389	3,196
all	122,397	3,346

The linguistic transformations above have been straightforward. However, we are also faced with lexicographical issues that require further consideration. The first of these concerns entries with multiple senses: homonymous and polysemous words. While the former needs no justification, the interpretation of polysemy, as well as the question of when it warrants multiple definitions, is much debated (see e.g., Bolinger 1965; Kirsner 1993 and the chapter on lexemes in Kornai 2018). Aside from any theoretical qualms one may have, there is also a practical one: even if the different senses of a word are numbered, its occurrences in the definitions are not, preventing us from effectively using this information. Therefore, we decided to merge the entries of multi-sense words by simply concatenating the definitions pertaining to the different senses.

The second problem is inconsistency. One would logically expect that each word used in a definition is itself defined in the dictionary; however, this is not the case. Such words should definitely be added to the ouroboros, but having no definition themselves, would contribute little to its semantics. As such, we eliminated them with an iterative procedure that also deleted entries whose definition became empty as a result. The procedure ran for 3–4 iterations, the number of removed entries/tokens ranging from 5342/912,373 on the raw dictionary to 707/125,509 on the most heavily filtered version.

Finally, in some entries, the definiens contains the definiendum. Since the presence or absence of these references – an artifact of the syntax of the language the definition is written in, not the semantics of the word in question – is arbitrary, they were removed as well.

2.2. The Ouroboros

Once the dictionary is ready, the next step towards the embedding is creating the Ouroboros set, which will serve as the basis of the word vector space. The Ouroboros is generated by an iterative algorithm that takes the definition graph as input and removes vertices at each iteration. The vertex set that remains at the end is the Ouroboros. A high-level pseudocode of the algorithm is included at the end of this section.

An iteration consists of two steps. In the first, we iterate through all words and select those that can be replaced by their definition. A word can be replaced if the following conditions hold:

1. no other words connected to the word in question in the definition graph (via both in- or out-edges) have been marked for replacement;

2. the vertex that corresponds to the word has no self-loop.

The first condition is simply a way of preventing race conditions in the replacement process. The second one, however, calls for some explanation. As we made sure that no definition contains its headword in the dictionary, initially, the definition graph contains no self-loops. However, as more and more words are removed, self loops start to appear. This is also our final stopping condition: the algorithm exits when all remaining vertices are connected to themselves. One can look at this condition as a way of saying that a word in the Ouroboros cannot be defined solely in terms of other words – in a way, it eats its own tail.

The second step performs the actual replacement. It removes the vertices marked by the first step, and connects all of their direct predecessors in the graph to their directs successors. In the weighted version, the weights are updated accordingly: the weight of a new edge will be equal to the product of the weights of the two edges it replaces. Figure 1 illustrates the replacement procedure with an example.

This step is also responsible for building the embedding graph. At first, the graph is empty. Each vertex removed by the algorithm, together with its in-edges, is added to it. By the time the algorithm stops, all vertices will have been added to the embedding graph. It is easy to see that this graph

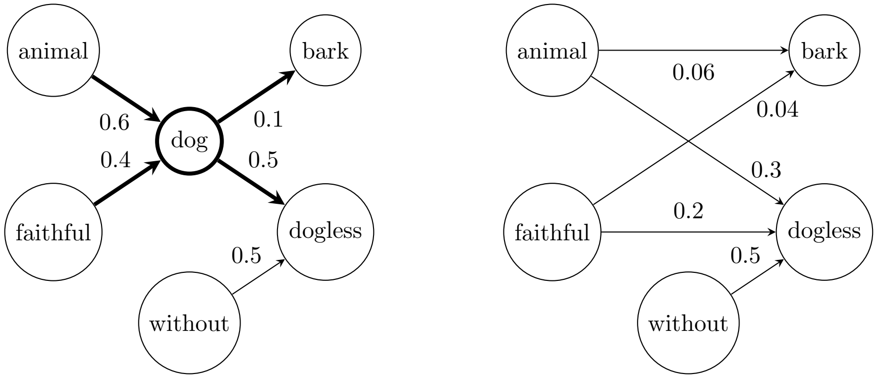


Figure 1: The definition graph before (left) and after (right) the word *dog* is replaced with its predecessors. Note that the sum of the in-weights for *bark* and *dogless* remains constant.

is a DAG, with the ouroboroi as its sources: what the whole algorithm effectively does is decrease the size of the cycles in the definition graph, vertex-by-vertex and edge-by-edge. The cycles never disappear completely, but become the self loops that mark the Ouroboros set. It follows then, that the Ouroboros contains at least one vertex and one edge (the self loop) from each cycle in the original definition graph and thus the embedding graph is free of (directed) cycles.

This algorithm can be tuned in several ways. The attentive reader might have noticed that the order in which the words are evaluated in the first step strongly determines which end up as replaceable. Several strategies were considered, including alphabetical and random order, shortest/longest, or rare/most common word (in definitions) first. Not surprisingly, rare words first performed best: this agrees with the intuition that the "basis" for the embedding should mostly contain basic words. Consequently, all numbers reported in this paper were attained with the rare first strategy.

We also experimented with decreasing the size of the embedding graph by deleting edges below a certain weight threshold; this is the equivalent of magnitude-based pruning methods in neural networks (Hertz et al. 1991). However, the performance of the embeddings created from pruned and unweighted graphs lagged behind those created from weighted ones. Hence, we used the latter for all experiments.

Algorithm 1 The Ouroboros algorithm

```
function CREATE_UROBOROS(dictionary)
    DG    CREATE_DEFINITION_GRAPH(dictionary)
    EG    Graph()
    repeat
        replaceable    COLLECT_REPLACEABLE(DG)
        if LENGTH(replaceable) > 0 then
            DO_REPLACE(DG, EG, replaceable)
    until LENGTH(replaceable) > 0
    return DG, EG
```

2.3. The embedding

This section describes the algorithm that takes as its input the ouroboros and embedding graphs and produces a word embedding. First, the basis of the vector space is computed. Since our goal is to describe all words in terms of the Ouroboros, the vector space will have as many dimensions (denoted with D) as there are vertices in the Ouroboros graph. Each coordinate corresponds to a word; the mapping is arbitrary, and we opted for alphabetic order. The word vectors for the ouroboroi (the first D rows of the embedding) are chosen to be the basis vectors of the vectors space.

The basis vector for an Ouroboros word w, however, can be calculated in two ways:

1. *Ouroboros-as-coordinates (OAC)*: as a sparse vector, where the only nonzero coordinate is the one that corresponds to the word itself. The first D rows of the embedding thus form the identity matrix.

2. *Ouroboros-as-vectors (OAV)*: as a vector whose nonzero coordinates correspond to the direct predecessors of w in the ouroboros graph. The values of the coordinates are the weights of the edges between its predecessors and w.

The two variants have opposing properties. OAV is much denser, which might bring words much closer in the semantic space than they really are, introducing "false semantic friends". OAC, on the other hand, is so sparse that the similarity of two Ouroboros words is always zero. This property might be useful if our algorithm was guaranteed to find the most semantically distributed feedback vertex set; however, no such guarantee exists. Since it is hard to choose between the two based solely on theoretical grounds, both variants are evaluated in the next chapter.

The vectors for the rest of the words are computed from the embedding graph. The graph is sorted topologically, with the ouroboroi at the beginning. The algorithm iterates through the words. The vector of a word w is set to be the weighted sum of the vectors of its direct predecessors in the embedding graph:

$$v_w = \sum_{w':(w',w)\in EG} v_{w'} \cdot e_{(w',w)},$$

where $e_{(i,j)}$ is the weight of the edge between i and j. The topological sort ensures that by the time we arrive to w, the vectors for all w's have already been calculated.

More by accident than design, we also created a third embedding beside OAC and OAV. Here the basis is taken from OAV, but the rest of the vectors are the same as for OAC; accordingly, we named it *Chimera (CHI)*. While the construction of this embedding is mathematically incorrect, it performed unexpectedly well, so we included it in the evaluation alonside OAC and OAV.

3. Evaluation

The algorithm presented in section 2 creates word embeddings, i.e., mappings from the vocabulary of a dictionary dataset to real-valued vectors of fixed dimension. This section will present two sets of experiments, both of which indicate that the distance between pairs of word vectors is a meaningful measure of the semantic similarity of words. In section 3.1 we will use two semantic similarity benchmarks for measuring semantic similarity of English word pairs to evaluate and compare our word embeddings. Section 3.2 presents a qualitative, manual analysis of each embedding that involves observing the set of words that are mapped to vectors in the immediate vicinity of a particular word vector in the embedding space.

3.1. Benchmark performance

The embeddings were evaluated on two benchmarks: `SimLex-999` (Hill et al. 2015) and `WS-353` (Finkelstein et al. 2002).

`SimLex` is the new standard benchmark for the task of measuring the semantic similarity of English word pairs. It contains 999 word pairs, each annotated with a gold standard similarity score, the average of scores given by human annotators. Performance of systems is measured as the Spearman correlation between a system's scores and the gold standard scores. State of the art systems achieving correlation scores in the 0.7–0.8 range (Mrkšić et al. 2016; Recski et al. 2016) combine multiple word embeddings and lexical resources, other competitive systems use word embeddings customized for the task of measuring word similarity (Schwartz et al. 2015; Wieting et al. 2015). General-purpose embeddings typically achieve a correlation in the 0.1–0.5 range; scores for some commonly used models are shown in Table 2.

The `WS-353` dataset contains 353 word pairs. It was originally devised to quantify any kind of semantic association: both similarity and *relatedness*. Here we use the subset that targets the former, selected by Agirre et al. (2009). Similarly to `Simlex`, performance is measured by Spearman's ρ. `WS-353` has been around longer than `Simlex`, and various corpus- (Gabrilovich & Markovitch 2007; Halawi et al. 2012) and knowledge-based methods (Hassan & Mihalcea 2011) have been evaluated against it; the current state-of-the-art, 0.828 was achieved by a hybrid system that also makes use of word embeddings (Speer et al. 2017).

Table 2: Coverage and performance of some word embeddings, measured by Spearman's ρ

System	Simlex		WS-353	
	Coverage	ρ	Coverage	ρ
Huang et al. (2012)[4]	996	0.14	196	0.67
`SENNA`[5] (Collobert & Weston 2008)	998	0.27	196	0.60
`GloVe.840B`[6] (Pennington et al. 2014)	999	0.40	203	0.80
`Word2Vec`[7] (Mikolov et al. 2013)	999	0.44	203	0.77

We evaluate various versions of our `ouroboros`-embeddings on both datasets. Results are presented in Table 3. Top scores on `Simlex` are just above 0.2, which outperforms Huang, but falls short of GloVe and Word2Vec by a similar margin. On the much easier `WS-353` dataset, even our best result is below that of the competition. Nevertheless, these results confirm that

[4] http://www.socher.org

[5] http://ronan.collobert.com/senna/

[6] https://nlp.stanford.edu/projects/glove/

[7] https://code.google.com/archive/p/word2vec/

our method yields vectors that are at least comparable to other general-purpose embeddings.

An early observation is that embeddings created using the OAV condition (see section 2.3) perform considerably worse than those built with the OAC condition. The most surprising part is the performance of the CHI embedding: while it tails behind the other two methods on `Simlex`, it improves dramatically when stopwords are filtered (the last two rows), to the extent that it becomes the best method on both datasets.

Table 3: Coverage and correlation of Wiktionary embeddings on `Simlex` and WS-353

Preprocessing	Simlex				WS-353			
	Cov.	ρ_{OAC}	ρ_{CHI}	ρ_{OAV}	Cov.	ρ_{OAC}	ρ_{CHI}	ρ_{OAV}
none	943	0.18	0.04	0.11	193	0.19	0.18	0.10
lowercasing	961	0.21	0.03	0.08	191	0.17	0.23	0.11
lemmatization	956	0.17	0.02	0.08	197	0.23	0.25	0.17
no multiword	943	0.15	0.03	0.10	193	0.19	0.15	0.08
no proper nouns	943	0.14	0.04	0.08	186	0.21	0.20	0.15
no punctuation	943	0.15	0.03	0.09	193	0.21	0.17	0.10
no stopwords	938	0.17	**0.22**	0.15	192	0.27	**0.46**	0.19
all	956	0.21	0.20	0.16	188	0.30	**0.46**	0.25

In order to gain further insight into how the three embeddings behave differently, we devised a further experiment based on the `all` embedding. The word pairs in the evaluation datasets have been divided into three groups, depending on how many of the two words are ouroboroi. Table 4 presents the results. Unsurprisingly, the numbers for CHI equal to OAV when both words are in the Ouroboros and to OAC when neither is. Perhaps predictably, our concerns about both OAC and OAV have been confirmed by the results: the orthogonal OAC basis breaks down when both words in a pair are in it, while the over-dense OAV fails to quantify the similarity of out-of-basis pairs. CHI, on the other hand, manages to be the "best of both worlds", at least as far as the first and the last row is concerned. Its exceptional performance in the middle row (in italics) is perplexing, because this is the point where OAV basis vectors are measured against OAC vectors; where the snake meets the lion, so to speak. Unfurling this mystery is left as future work.

Table 4: A more in-depth look into the performance of the `all` embedding

Word in basis	Simlex				WS-353			
	Size	ρ_{OAC}	ρ_{CHI}	ρ_{OAV}	Size	ρ_{OAC}	ρ_{CHI}	ρ_{OAV}
Both	313	0.00	0.13	0.13	46	0.00	0.27	0.27
One	468	0.27	*0.20*	0.21	85	0.34	*0.53*	0.36
Neither	175	0.30	0.30	0.12	57	0.50	0.50	0.23

Both `SimLex` and WS-353 contain pairs of frequent words. Our hope is that in the next section our method will show its strength on infrequent words that cause trouble for distributional models that are limited by the amount of training data available.

3.2. Nearest neighbors

As mentioned in section 1, we expect our embeddings to yield meaningful representations even for infrequent words that pose a problem for distributional approaches. We have no knowledge of reliable datasets containing the semantic similarity of infrequent words, a quantitative analysis is therefore not possible. A more subjective method to evaluate whether the angle between word vectors is proportional to semantic similarity is to observe vectors in the immediate vicinity of a particular vector to see whether they are semantically related to the word corresponding to that vector. Our experiment involves examining the nearest neighbors of vectors corresponding to a small sample of infrequent words in our least noisy `ouroboros`-embedding (using all filtering steps on the Wiktionary data) and a large, publicly available embedding trained using GloVe on 840 billion words of raw English text and containing vectors for 2.2 million words.

To create a sample of infrequent English words, we used a word frequency list constructed from the UMBC Webbase Corpus (Han et al. 2013). To extract words that are in English, correctly spelled, and can be expected to appear in a dictionary, we matched the list against the full vocabulary used in a late draft version of (Kornai 2018), which we know to contain many infrequent words. After manually excluding from the resulting list technical words related to mathematics or linguistics, we kept the five least frequent ones for the purposes of the current experiment. The five words, along with their definitions in Wiktionary, are shown in Table 5. For both the uroboros and GloVe embeddings we extracted the nearest neighbors of each of the five words in our sample. Tables 6 and 7 show for each word the top two neigbors in the `uroboros` and `GloVe` embeddings, respectively. We also include Wiktionary definitions of these neighbor words, where available.

Table 5: Sample of five infrequent words used in (Kornai 2018)

Word	Wiktionary definition
compter	A counter (token used for keeping count)
	A prison attached to a city court; a counter
entelechy	The complete realisation and final form of some potential concept or function
	A particular type of motivation, need for self-determination,
	and inner strength directing life and growth to become all one is capable of being
hinny	The hybrid offspring of a stallion (male horse) and a she-ass (female donkey).
perron	A stone block used as the base of a monument, marker, etc.
	A platform outside the raised entrance to a church or large building
quodlibet	A form of music with melodies in counterpoint.
	A form of trompe l'oeil which realistically renders domestic items

Table 6: Nearest neighbors of our sample words in the `ouroboros` embedding

Word	Neighbor	Definition
compter	jeton	a counter or token
	countify	to use as a count noun
entelechy	subtyping	a form of type polymorphism (...)
	convolve	to compute the convolution function
hinny	fummel	a hinny
	zebrinny	the offspring of a male horse and a female zebra
perron	stereobate	the foundation, typically of a stone building
		the steps of the platform beneath the stylobate
	jamo	any of the 24 building blocks of the Korean (hangeul) alphabet.
quodlibet	planctus	a lament or dirge, a popular literary form in the Middle Ages.
	chorale	a chorus or choir.
		a form of Lutheran or Protestant hymn tune.

Table 7: Nearest neighbors of our sample words in the `GloVe` embedding

Word	Neighbor	Definition
compter	compuer	n/a
	copouter	n/a
entelechy	aristotelianism	the philosophical system of Aristotle and his followers
	somethingness	the quality of being something
hinny	tuchus	alternative form of *toches* → the buttocks, rear end, butt
	hiney	buttocks
perron	chingon	(as *chingón:*) (Mexico, slang) very smart, intelligent (...)
	chido	(Mexico, slang) cool, acceptable, easy
quodlibet	sequitur	A logical conclusion or consequence of facts.
	peric	n/a

Even such a small and non-representative sample of infrequent English words is sufficient to exemplify some of the issues that arise when representing infrequent words with distributional models. Typos of more frequent words may dominate the total number of occurrences in a corpora: *compter* and *hinny* are clearly represented by the `GloVe` embedding as alternative forms of *computer* and *hiney*, respectively. Neighbors of the other three sample words in the `GloVe` embedding are seemingly random. Meanwhile, in 4 out of the 5 example cases, `uroboros` maps rare words into the vicinity of highly related lexemes.

4. Conclusion

In this work, we examined the possibility of creating word embeddings from a dictionary. While the performance of our embedding in the word similarity task lags behind those obtained by prediction-based methods, it is perhaps better suited to find relevant neighbors of rare words.

In future work, we hope to iron out the sparsity/density problem that is, in part, responsible for the lackluster similarity scores. Another avenue of research we intend to pursue is to consolidate prediction- and dictionary-based embeddings into a hybrid model that combines the advantages of both.

Acknowledgements

Research partially supported by National Research, Development and Innovation Office NKFIH grant #120145 and by National Excellence Programme 2018-1.2.1-NKP-00008: Exploring the Mathematical Foundations of Artificial Intelligence.

References

Agirre, E., E. Alfonseca, K. Hall, J. Kravalova, M. Paşca and A. Soroa. 2009. A study on similarity and relatedness using distributional and Wordnet-based approaches. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 19–27.

Banarescu, L., C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer and N. Schneider. 2013. Abstract meaning representation for sembanking. In Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse. Sofia: Association for Computational Linguistics, 178–186.

Bird, S., E. Klein and E. Loper. 2009. Natural language processing with Python. Sebastopol, CA: O'Reilly Media.

Bolinger, D. L. 1965. The atomization of meaning. Language 41. 555–573.

Bullon, S. (ed.). 2003. Longman dictionary of Contemporary English 4. Harlow: Longman.

Collobert, R. and J. Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference on Machine Learning. New York: ACM, 160–167.

Finkelstein, L., E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. 2002. Placing search in context: The concept revisited. ACM Transactions on Information Systems 20. 116–131.

Gabrilovich, E. and S. Markovitch. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In IJcAI. 1606–1611.

Halawi, G., G. Dror, E. Gabrilovich and Y. Koren. 2012. Large-scale learning of word relatedness with constraints. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 1406–1414.

Han, L., A. L. Kashyap, T. Finin, J. Mayfield and J. Weese. 2013. UMBC_EBIQUITY-CORE: Semantic textual similarity systems. In Second Joint Conference on Lexical and Computational Semantics (*SEM). 44–52.

Hassan, S. and R. Mihalcea. 2011. Semantic relatedness using salient semantic analysis. In AAAI.

Hertz, J. A., A. S. Krogh and R. G. Palmer. 1991. Introduction to the theory of neural computation. Redwood City, CA: Addison-Wesley.

Hill, F., K. Cho, A. Korhonen and Y. Bengio. 2016. Learning to understand phrases by embedding the dictionary. Transactions of the Association for Computational Linguistics 4. 17–30.

Hill, F., R. Reichart and A. Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. Computational Linguistics 41. 665–695.

Huang, E., R. Socher, C. Manning and A. Ng. 2012. Improving word representations via global context and multiple word prototypes. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012). Jeju Island, Korea: Association for Computational Linguistics. 873–882.

Kirsner, R. S. 1993. From meaning to message in two theories: Cognitive and Saussurean views of the Modern Dutch demonstratives. Conceptualizations and mental processing in language. 80–114.

Kornai, A. 2010. The algebra of lexical semantics. In C. Ebert, G. Jäger and J. Michaelis (eds.) The mathematics of language. 10th and 11th Biennial Conference, MOL 10, Los Angeles, CA, USA, July 28–30, 2007 and MOL 11, Bielefeld, Germany, August 20–21, 2009, Revised Selected Papers. Berlin & Heidelberg: Springer. 174–199.

Kornai, A. 2012. Eliminating ditransitives. In P. de Groote and M.-J. Nederhof (eds.) Revised and selected papers from the 15th and 16th Formal Grammar Conferences. Berlin: Springer, LNCS 7395. 243–261.

Kornai, A. 2018. Semantics. Berlin: Springer.

Kornai, A., J. Ács, M. Makrai, D. M. Nemeskey, K. Pajkossy and G. Recski. 2015. Competence in lexical semantics. In Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics (*SEM 2015). Denver: Association for Computational Linguistics. 165–175.

Manning, C. D., M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard and D. McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In Association for Computational Linguistics (ACL) system demonstrations. 55–60.

Mikolov, T., K. Chen, G. Corrado and J. Dean. 2013. Efficient estimation of word representations in vector space. In Y. Bengio and Y. LeCun (eds.) Workshop proceedings, International Conference on Learning Representations (ICLR 2013).

Miller, G. A. 1995. WordNet: A lexical database for English. Communications of the ACM 11. 39–41.

Mrkšić, N., D. Ó Séaghdha, B. Thomson, M. Gašić, L. M. Rojas-Barahona, P.-H. Su, D. Vandyke, T.-H. Wen and S. Young. 2016. Counter-fitting word vectors to linguistic constraints. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego, CA: Association for Computational Linguistics, 142–148.

Pennington, J., R. Socher and C. Manning. 2014. Glove: Global vectors for word representation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014).

Quillian, M. R. 1968. Word concepts: A theory and simulation of some basic semantic capabilities. Behavioral Science 12. 410–430.

Quine, W. v. O. 1951. Two dogmas of empiricism. The Philosophical Review 60. 20–43.

Recski, G. 2016. Building concept graphs from monolingual dictionary entries. In N. Calzolari, K. Choukri, T. Declerck, M. Grobelnik, B. Maegaard, J. Mariani, A. Moreno, J. Odijk and S. Piperidis (eds.) Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). Portorož: European Language Resources Association (ELRA).

Recski, G. 2018. Building concept definitions from explanatory dictionaries. International Journal of Lexicography 31. 274–311.

Recski, G., E. Iklódi, K. Pajkossy and A. Kornai. 2016. Measuring semantic similarity of words using concept networks. In Proceedings of the 1st Workshop on Representation Learning for NLP. Berlin: Association for Computational Linguistics, 193–200.

Schwartz, R., R. Reichart and A. Rappoport. 2015. Symmetric pattern based word embeddings for improved word similarity prediction. In Proceedings of the 19th Conference on Computational Natural Language Learning (CoNLL 2015). Beijing: Association for Computational Linguistics, 258–267.

Speer, R., J. Chin and C. Havasi. 2017. ConceptNet 5.5: An open multilingual graph of general knowledge. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17). 4444–4451.

Wieting, J., M. Bansal, K. Gimpel, K. Livescu and D. Roth. 2015. From paraphrase database to compositional paraphrase model and back. TACL 3. 345–358.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 19–24

On binding relations

ASH ASUDEH
Oxford University & Carleton University
ash.asudeh@ling-phil.ox.ac.uk

KEYWORDS

binding theory
binding relations
reflexives
coindexation
linking
Lexical-Functional Grammar

ABSTRACT

A central concern in the syntax of pronominals is the correct representation of the syntactic relationship between a pronominal and its antecedent, where we can think of the antecedent as the expression that determines the pronominal's interpretation. Elements like reflexives are of particular interest, because they must have syntactically circumscribed antecedents, in contrast to pronouns, which may refer deictically. Two common ways to represent the binding relation between a reflexive and its antecedent are through coindexation or linking, where these are seen as strict alternatives to each other. Coindexation is symmetric and transitive, whereas linking is asymmetric and intransitive. However, this raises a problem, as empirical data has shown that both transitivity and asymmetry are required of binding relations. A solution presents itself in the binding equations of Lexical-Functional Grammar, which are transitive due to their use of equality (a standard transitive relation), but asymmetric due to their use of an ANTECEDENT feature (if *x* is the ANTECEDENT of *y*, *y* is not the ANTECEDENT of *x*).

One common way to represent the relationship between a pronominal and its antecedent – let's call it "the coconstrual relation" (following Safir 2004a;b) – is through the use of indices on nominals:

(1) Alli_1 told Thora_2 that $\text{she}_{1/2/3}$ was next.

The proper names *Alli* and *Thora* have distinct indices, which is understood to mean that they refer to distinct individuals. The pronoun *she* can be coindexed with either of the proper names, in which case it refers to the same individual as the name in question. The indexing relation is transitive such that if *Alli* and *she* are coindexed, then *Thora* and *she* are contra-indexed, since *Alli* and *Thora* are contra-indexed. Lastly, the pronoun need not be coindexed with either of the names mentioned, in which case it bears an index contra-indexed with both names.

Coindexation is a *symmetric* relation: if A is coindexed with B, then B is coindexed with A. A variety of *asymmetric* representations of the antecedent–pronominal relation have also been explored (Higginbotham 1983; 1985; Dalrymple 1993; Heim 1998; Fox 2000; Safir 2004a;b; Büring 2005a;b). As Higginbotham (1983) points out, only an asymmetric relation actually directly captures the antecedence relation. This is evident if we compare the coindexation representation in (2) to Higginbotham's *linking* representation in (3), where the head of the arrow is at the antecedent and the tail of the arrow is at the anteceded element:

(2) Thora$_1$ said she$_1$ thought Alfred had tickled her$_1$.

(3) a. Thora said she thought Alfred had tickled her.

b. Thora said she thought Alfred had tickled her.

The arrow notation makes it clear that the linking relation is asymmetric and represents that *Thora* is the antecedent of *she* and that *she* is the antecedent of *her* in (3b), whereas *Thora* is the antecedent of both *she* and *her* in (3a). In contrast, the coindexation in (2) does not capture whether *Thora* or *she* is the antecedent of *her*.[1] Heim (1998) proposes a notational variant of the linking arrows, using dual indices (which she calls "inner" and "outer" indices); Büring (2005a;b) also adopts a kind of dual indexation.[2]

[1] Coindexation per se does not even capture, e.g., whether *Thora* is the antecedent of *she* or vice versa, but independent considerations in any theory that uses coindexation would settle this question in favour of the former option.

[2] It should be noted that both the coindexation and linking syntactic representations in fact represent two different kinds of semantic relationship between the pronoun and its antecedent, where the exact nature of the relationship depends on the nature of the antecedent. If the antecedent is a referential noun phrase, the relationship can be one of simple *coreference*. If the antecedent is an operator, the relationship must be something akin to logical variable binding (Bach & Partee 1980; Büring 2005a, 81ff). The following standard sort of example makes this clear; for simplicity, let us assume that the others in question are not related to Harry:

(i) Only Harry heard his sister.
Interpretation 1: Only Harry is an x who heard Harry's sister.
Interpretation 2: Only Harry is an x who heard x's sister.

The first interpretation is a coreferential interpretation, such that the pronoun refers to whatever the proper name *Harry* refers to. On this interpretation, the others did not hear Harry's sister, but may have heard their own sisters. The second interpre-

These representational differences in binding relations have linguistic consequences, although this is not obvious from (2) and (3) alone. Let us call the configuration in (3a) *cobinding* and the configuration in (3b) *transitive binding*, following Heim (1998) and Büring (2005b). The following example from Büring (2005b, 264) – modified to use the linking representation – illustrates that cobinding and transitive binding can give rise to distinct interpretations (all caps indicates focus):

(4) Every man is afraid that only HE voted for his proposal.

a. Cobinding:
every man is afraid that only he voted for his proposal

Fear: 'No one else voted for my proposal!'

b. Transitive binding:
every man is afraid that only he voted for his proposal

Fear: 'No one else voted for his own proposal!'

A symmetric relation like coindexation cannot capture the distinction between cobinding and transitive binding, making it difficult for a theory that represents the antecedent–pronominal relation symmetrically to account for (4).

Another distinction between the coindexation relation and the linking relation is that the former is *transitive*, whereas the second is not. The issue of transitivity is a long-standing one in the syntax of pronominals. In the early literature on pronominal syntax (Jackendoff 1972; Wasow 1972), the problem concerned how to relate the reflexive to the matrix subject in (5) without introducing a rule that would also overgenerate (6). From this point on, I use bold face in example sentences to indicate coconstrual.

(5) **Thora** worried that **she** might implicate **herself**.

(6) ***Thora** worried that Alfred might implicate **herself**.

Coindexation plus a locality restriction on the antecedent–anaphor relation neatly solved this problem: the reflexive must have a suitable local antecedent. *Alfred* and *she* are both local, but only *she* is a suitable antecedent for the feminine reflexive (assuming *Alfred* names a male individual, as is

tation is the semantically bound interpretation, such that the pronoun is bound by the quantificational noun phrase *Only Harry*. On this interpretation, the others in question did not hear their own sisters, but may have heard Harry's sister.

conventionally the case). If the reflexive is coindexed with *she*, it would be related to *Thora* in (5) by transitivity of coindexation, on the relevant reading where *Thora* is coindexed with *she*. However, it cannot be directly related to *Thora*, as would have to be the case for (6) to be grammatical, because the matrix subject is not sufficiently local to the reflexive.

The transitivity of coindexation similarly explains the pattern in (7), which turns out to be a problem for asymmetric relations such as linking:

(7) a. ***Isak** said **he** saw **him**.
b. **Isak** said **he** saw him.
c. **Isak** said he saw **him**.

Isak can be the antecedent of *he* or *him*, as in (7b) and (7c), but it cannot be the antecedent of both, as in (7a). On the standard assumption that these pronouns can only take antecedents that are suitably non-local, if *he* and *him* are both coindexed with *Isak*, then *he* and *him* are also coindexed with each other, but *he* is too local to *him*. Lasnik (1976) discusses similar examples, in light of the previous work on transitivity of coconstrual mentioned above.

Higginbotham (1983, 406) immediately observed that (7a) is problematic for an asymmetric antecedent–pronominal relation, because there is a cobinding representation in which *he* is not the antecedent of *him*, such that each of *he* and *him* have the relation to *Isak* that they have in (7b) and (7c), which are independently grammatical:

(8) Isak said he saw him.

Büring (2005b, 265) points out that these sorts of examples have normally lead to various theoretical complications for asymmetric theories of coconstrual.

However, it is possible to simultaneously reap the benefits of asymmetric linking and transitive coindexation through an antecedent–pronominal relation that is *both asymmetric* and *transitive*. One such relation is that of Dalrymple (1993), which is couched in the theory of Lexical-Functional Grammar (Kaplan & Bresnan 1982; Bresnan et al. 2016; Dalrymple 2001).

Dalrymple's relation can be abbreviated as follows, where f can be thought of as the pronominal in question:[3]

(9) $(f\ \text{ANTECEDENT})_\sigma = f_\sigma$

This constraint states that – at the level of *semantic structure* (Halvorsen & Kaplan 1988; Dalrymple 1999; 2001; Asudeh 2012), indicated by the subscript σ – the pronominal is equal to its antecedent. The feature ANTECEDENT introduces *asymmetry*: it is not the case that if A is the ANTECEDENT of B, then B is the ANTECEDENT of A. Equality introduces *transitivity*: if A is the antecedent of B and C, then (at semantic structure) A equals B and A equals C, which means that B equals C.

This relation captures the distinction between cobinding and transitive binding, as in (4), due to the asymmetry of the antecedent–anaphor relation. It also correctly captures the pattern in (7) while correctly ruling out (8), due to the transitivity of equality.

References

Asudeh, A. 2012. The logic of pronominal resumption. Oxford: Oxford University Press.

Bach, E. and B. H. Partee. 1980. Anaphora and semantic structure. In J. Kreiman and A. E. Ojeda (eds.) Papers from the Parasession on Pronouns and Anaphora. Chicago: Chicago Linguistic Society. 1–28.

Bresnan, J., A. Asudeh, I. Toivonen and S. Wechsler. 2016. Lexical-Functional Syntax (2nd ed.). Malden, MA & Oxford: Wiley-Blackwell.

Büring, D. 2005a. Binding theory. Cambridge: Cambridge University Press.

Büring, D. 2005b. Bound to bind. Linguistic Inquiry 36. 259–274.

Dalrymple, M. 1993. The syntax of anaphoric binding. Stanford: CSLI.

Dalrymple, M. (ed.). 1999. Semantics and syntax in Lexical Functional Grammar: The resource logic approach. Cambridge, MA: MIT Press.

Dalrymple, M. 2001. Lexical Functional Grammar (Syntax and semantics 34). New York: Academic Press.

Fox, D. 2000. Economy and semantic interpretation. Cambridge, MA: MIT Press.

Halvorsen, P.-K. and R. M. Kaplan. 1988. Projections and semantic description in Lexical-Functional Grammar. In Proceedings of the International Conference On Fifth Generation Computer Systems. Tokyo, 1116–1122.

Heim, I. 1998. Anaphora and semantic interpretation: A reinterpretation of Reinhart's approach. In U. Sauerland and O. Percus (eds.) The interpretive tract (MIT Working Papers in Linguistics 25). Cambridge, MA: MIT Press. 205–246.

[3] This abbreviation also captures the effect of the basic Glue Semantics (Dalrymple 1999) treatment of anaphora, which treats the pronominal as a function on its antecedent. See Asudeh (2012) for further details.

Higginbotham, J. 1983. Logical form, binding, and nominals. Linguistic Inquiry 14. 395–420.

Higginbotham, J. 1985. On semantics. Linguistic Inquiry 16. 547–593.

Jackendoff, R. 1972. Semantic interpretation in generative grammar. Cambridge, MA: MIT Press.

Kaplan, R. M. and J. Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In J. Bresnan (ed.) The mental representation of grammatical relations. Cambridge, MA: MIT Press. 173–281.

Lasnik, H. 1976. Remarks on coreference. Linguistic Analysis 2. 1–22.

Safir, K. 2004a. The syntax of anaphora. Oxford: Oxford University Press.

Safir, K. 2004b. The syntax of (in)dependence. Cambridge, MA: MIT Press.

Wasow, T. 1972. Anaphoric relations in English. Doctoral dissertation. MIT.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 25–57

Variation in the nominal morphology of Northern Vlax Romani

MÁRTON ANDRÁS BALÓ
Research Institute for Linguistics,
Hungarian Academy of Sciences
balo@nytud.hu

KEYWORDS

variation
analogy
patterns
Construction Morphology
Vlax Romani

ABSTRACT

This paper discusses two instances of variation in the nominal morphology of Northern Vlax Romani varieties as spoken in Hungary. The discussion is conducted in an analogical framework, relying only on surface forms and their relationships, using Construction Morphology and taking the notion of schemata as introduced by Booij (2010) one step further. I will also make an attempt at defining the notion of a weak point as a locus of the emergence of variation. If different morphological tools are employed by different stems for the same semantic function within a strictly delimited paradigm, pattern-seeking may begin. The different patterns may result in one and the same stem employing different tools to express the same function; thus, the patterns will serve as different analogical forces that influence the extent and nature of variation. This paper focuses on variation in the strict sense, that is, phenomena involving vacillating stems.

1. The data

Romani is the only New Indo-Aryan language not spoken in India but rather in Europe, its closest relatives among other Indo-Aryan languages being Rajasthani and Gujarati. Although realistic estimates of the number of speakers are not easy to make, Bakker (2001) put their number at approximately 4.6 million in Europe at the beginning of the third millennium. The total number of the Romani people in Europe, including both speakers and non-speakers, has been estimated at anywhere between 4 and 12 million. Due to more recent migrations, Romani has also been spoken in the Americas, where the numbers are even harder to determine, but conservative estimates (cf. Matras 2005) suggest that there are upwards of 500 000 speakers, and there are probably more, as there are about 800 000 Romani people living in Brazil alone (cf. Gaspar 2012) and approximately one million in the United States (Hancock 2013). Romani monolingualism virtually does not exist; all speakers of Romani are at least bilingual.

The dialect classification still in use in current Romani linguistic literature builds upon the branches established by Miklosich (1872–1880), relying mostly on contact phenomena. One of the four main dialect groups of Romani is Vlax Romani, which is, based on certain diagnostic features, further divided into a Southern and a Northern group, with the former spoken mostly in the Balkans, while the latter in Romania, Hungary, Moldova and Serbia. Some confusion is caused by the fact that the Northern Vlax group has often been referred to simply as Vlax Romani in papers written about Romani as spoken in Hungary (e.g., Erdős 1959; Vekerdi 2000).

Authentic and trustworthy corpora as such, of any variety of Romani, have not existed until very recently, and when looking into instances of synchronic variation, new and authentic data are of utmost importance. The situation in the international landscape is better now, although the small corpora of Thrace Romani–Turkish–Greek and Finnish Romani–Finnish have been collected with the aim of research into language contact, and the corpus of Russian Romani does not include the newly collected spoken data yet (Kozhanov 2016).

Therefore, we set out to collect new, authentic, up-to-date, real life Romani data in Hungary in 2015 within the framework of the project *Variation in Romani Morphology* (OTKA K 111961, PI: László Kálmán).[1] Based on a questionnaire specifically designed for this purpose by the author, but also recording spontaneous speech, we carried out fieldwork in several locations in Hungary, and thus far we have carried out Northern Vlax Romani interviews with 30 informants or groups of informants all over the territory of Hungary. Although the quantity of the data is not large, small corpora have actually been used effectively in the course of conducting valuable research (Adamou 2016). In the present study, we will exclusively rely on these data, occasionally referring to another fairly reliable but slightly outdated source, Vekerdi (1985). The Northern Vlax Romani varieties where the data come from include Lovari, Mašari and Drizari. Although sometimes considered as separate linguistic groups within Northern Vlax Romani (Erdős 1959; Tálos 2001), a comprehensive study on the Vlax dialects of Romani, Boretzky (2003) does not include them as separate dialects. Based on their similarity seen so far, I will consider them as one variety from a linguistic aspect, used by different, self-designated groups.

[1] The fieldwork was carried out by Mátyás Rosenberg and the author.

2. The notion of a weak point

In order to clarify what a weak point is,[2] I will use the idea that the regularities on a particular level of linguistic description can be expressed in terms of schemata (Booij 2010, following the notion of schema, as described by Rumelhart 1980). Although related, schemata represent a more complex notion than constructions. While the latter denote a pairing of form and meaning (Goldberg 1995; Jackendoff 2008), the former, in the case of morphological schemata, contains phonological, syntactic and semantic information.[3] For example, the schema for deverbal *-er* in English is as follows, where the symbol $\leftrightarrow$ stands for correspondence (Booij 2010, 8).

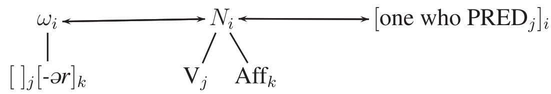


Figure 1: Schema for deverbal *-er* in English

The three kinds of linguistic information included here are the phonological form ω, the syntactic information, and the semantic information. The syntactic information in the original form of the schema is encoded as N, meaning that the word containing the deverbal *-er* suffix is a noun. There may often be a need for morpho-syntactic properties to be specified, however (Booij 2010, 7). As the precise elaboration of the syntactic component is not part of the present paper, I will use the more general symbol s to indicate that a schema like this can represent constructions

[2] A weak point is fundamentally similar to an unstable point, as defined by Rebrus and Törkenczy (2011). They define an unstable point in paradigms as "those points in the paradigm where more than one conflicting analogical requirement applies with approximately equal strength" (*op.cit.*, 139). Although the present paper will mainly deal with formal connections, they add that a functional relationship can also serve as an analogical connection.

[3] Rebrus and Törkenczy (2005) do something similar when they underspecify the input in the framework of Optimality Theory by defining its morpho-syntactic characteristics only and rely on output-output constraints to determine the outcome of two cases of lexical allomorphy in the Hungarian verbal paradigm. The two cases are Definiteness Neutralisation and Anti-Harmony, and the constraints they use require paradigmatic uniformity on the one hand and paradigmatic contrast on the other. We may say that, in some way, the underspecified inputs correspond to the semantic and the morpho-syntactic component of the schemata, while the correspondences between the components of a schema or between components of different schemata are similar to the ranking of the constraints.

on any level of morphological or syntactic complexity. Thus, the schema for the Hungarian plural suffix -*k* would be the one shown in Figure 2.

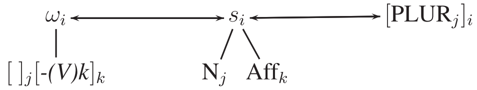


Figure 2: Schema for the Hungarian plural suffix -*k*

Instead of this linear representation, based on the idea of Booij (2010), I suggest a circular representation of the schema, as sketched in Figure 3, where every kind of information is connected to the other two through correspondences, marked by arrows in both directions, as there is also a relationship between the semantic and the phonological information. This is not unlike what Jackendoff (2012) suggests, for example, when he claims that a word like *cow* is stored in memory, and "it involves a pronunciation linked with a meaning and the syntactic feature *Noun*" (*op.cit.*, 176). One reason why it is important to postulate interrelations among all three components is, as it is pointed out by Jackendoff (2012), that there are words which lack one of the components, like *ouch*, which has phonology and meaning but lacks syntactic features. Another reason for postulating a direct relationship between the phonological and the semantic components is that it is a significant one in the argumentation below as the variation seen in the nominal morphology of Northern Vlax Romani takes place along the correspondence between these two components, while leaving the syntactic component intact, which provides evidence for the solution proposed here, a schema showing an interrelated matrix of the three components.

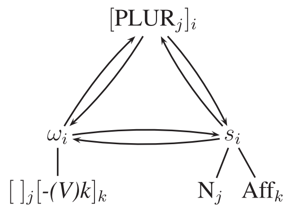


Figure 3: Improved schema for the Hungarian plural suffix -*k*

A schema like this becomes weaker when there is a disturbance in any of the correspondences. For example, if a new phonological form, ω_i' started to appear in the same syntactic position and with the same meaning as the deverbal -*er* or the plural -*k*, then this would weaken the schema, which may in turn trigger variation and the schema would become a weak point. It is also possible that more than one correspondence becomes unstable, like the locative case in Northern Vlax Romani, where the semantic component may pair up with a different phonological form and a different syntactic position, resulting in variation.[4] Thus, a weak point in morphology is a schema where at least one of the correspondences is not mutually unambiguous.

We can draw up the following, combined schema, shown in Figure 4, consisting of two schemata, for the locative case in Northern Vlax Romani. The upper section of the schema describes the agglutinative case marking: for example *e kheréste* 'in the house', where *e* is the inflected form of the definite article and *kheréste* is the locative form of *kher* 'house'. It contains the phonological form; the morpho-syntactic information, which says that the case affix is attached to the oblique base of the noun; and the semantic component, which is the locative function in this case. There is an alternative way of expressing the locative, by means of a preposition, shown in the lower section of the schema: *andó kher* 'in the house'. In addition to the noun *kher* 'house', this form is composed of the preposition *ande* 'in' and the base form *o* of the definite article.

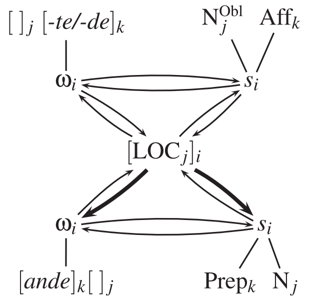


Figure 4: Schema for the locative case in Northern Vlax Romani

4 The variation investigated in the present paper is not completely unlike microvariation, but Barbiers et al. (2007) focus on inter-dialectal variation, using a typological or a generative approach. Further research into microvariation tries to account more "for the range and (limits) of inter- and intra-speaker variation in a principled way while at the same time testing existing formal theories against these microvariational data and thus contributing to the theory of language variation", but it still investigates closely related language variants, thus going with a dialectologically oriented approach, "applying the formal theoretical concepts of generative grammar" (Brandner 2012, 113). However, the variation we are dealing with in Northern Vlax Romani is intra-dialectal variation, which manifests itself as either inter- or intra-speaker variation.

The thick arrows in this schema mean that the correspondences in that direction prevail in the expression of the locative case, so the prepositional form is more typical than the agglutinative one. However, the presence of both forms suggests that the locative function does not exclusively correspond to either the form represented by agglutinative case marking or the form represented by the preposition.

As another example, let us take the English past tense. There is a strong relationship between the semantic function "past tense" and the way of marking commonly called "regular" (the addition of the suffix *-ed*). If all English verbs were inflected that way, there would only be one single schema for the past tense.

However, this is not the case. There are several alternative, so-called "irregular" verbs of lower or higher frequency, making up smaller or bigger groups (*sing–sang, cut–cut, keep–kept* etc.). The existence of these groups of verbs means that the correspondence between the past tense function and the marker *-ed* is not unambiguous, and neither is the correspondence between the past tense function and the morpho-syntactic property of affixation for the past tense. Several other morpho-syntactic ways and phonological forms are used in the formation of the English past tense, for example ablaut (*sing–sang*), vowel shortening (*keep–kept*) or reverse umlaut (*think–thought*).

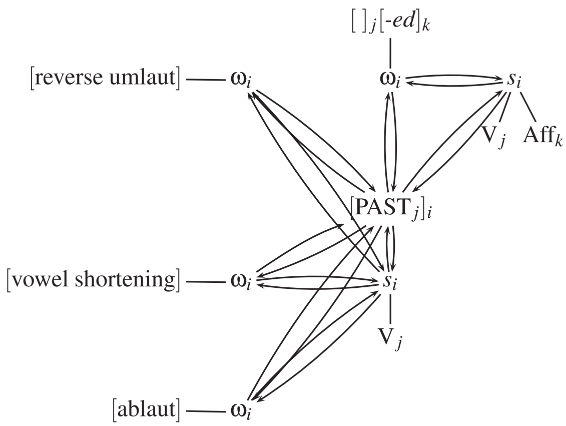


Figure 5: Schema for the English past tense

With so many schemata coalescing around the same semantic component, the correspondences become ambiguous and represent a weak point, where variation may emerge, although it does not necessarily do so. This probably depends on other factors, such as frequency, the extent of the embedded nature of the forms etc. However, if variation emerges, then we have every reason to think that there are patterns which are competing for the same function, or patterns which have some other kind of phonological or morpho-syntactic influence on the forms that begin to vary.

3. An overview of the weak points under discussion in Northern Vlax Romani

I will briefly introduce the two weak points in the nominal inflection of Northern Vlax Romani where variation occurs[5] and where the surface forms (surface similarities and differences; in general, cf. e.g., Kálmán et al. 2012) and analogical effects might play a role in producing and maintaining this variation.

1. The first weak point we will look at is the masculine oblique base.[6] One oblique marker for masculine nouns is *-es-* in the singular and *-en-* in the plural, so the oblique bases of a word like *šēró* 'head'[7] are *šērés-* and *šērén-*, respectively. However, this schema does not exclusively prevail within the masculine nouns. It is weakened by the existence of another phonological form, containing *-os-* in the singular and *-on-* in the plural, so, for example, the oblique forms of the word *hī́ro* 'a piece of news' are *hīrós-* and *hīrón-*, respectively.

2. The second weak point can be found in the feminine plural oblique base. The oblique marker in the singular is invariably *-a-*: *šej* 'girl'∼ *šejá-, žuv* 'louse'∼ *žuvá-*. However, there are two available patterns in the plural. One of the possible feminine plural oblique markers is *-an-*, for example the plural oblique base of *šej* 'girl' is *šeján-*; but there is also another phonological form of the feminine plural oblique marker, *-en-*, see for example *žuv* 'louse', whose plural oblique base is *žuvén-*.

[5] We must note that the present paper does not deal with the possible diachronic processes that could have led to this variation and are emphasised heavily in the literature on Romani linguistics.

[6] The case declension system of Romani primarily relies on a nominative/oblique opposition. Further case markers are attached to the oblique base (Matras 2002, 43–44), as is not unusual in New Indo-Aryan languages (Masica 1991, 230–248).

[7] Stress is marked by an acute accent, while a macron marks a long vowel.

4. The masculine oblique base

In this section, we will look at the first weak point, the masculine oblique base, in more detail. Following the description of the phenomenon in question, we will analyse two possible reasons for the weakness and the ensuing variation, and discuss to what extent there can be interaction between the possible reasons and the variation. They are the following.

1. The position of stress. At first glance, it seems that there is at least some sort of correlation between the variation of the oblique forms and the fact that Northern Vlax Romani lacks a straightforward stress pattern. Stress is lexical, which means it cannot be predicted based merely on the form of the word or the number of syllables. So, while the level of unpredictability is high by default, it is further complicated by the fact that the position of stress in certain words with three syllables may also vary. While the stress pattern of disyllabic words (word-initial or word-final) seems to determine the form of the oblique base unambiguously, the varying stress pattern of trisyllabic words pairs up with the unpredictability of oblique forms.
2. The number of syllables. This is related to the position of stress to some degree, as oblique forms begin to vary when the number of syllables reaches or exceeds three. The variation is especially ostensible on trisyllabic words with a stem-final /o/, while disyllabic words never vary.

4.1. Description of the phenomenon

In this section, we will introduce the variation in the masculine oblique base and we will also see that this variation is closely linked to the masculine nouns which have a stem-final /o/.

In Northern Vlax Romani, there are two sets of suffixes for the oblique base. One of them comprises *-es-* for the singular and *-en-* for the plural, but there are masculine nouns which, without any apparent phonological or morpho-phonological reason, take a different oblique marker: *-os-* in the singular and *-on-* in the plural.

(1) *šēró* 'head' → obl. *šērés-/šērén-*
hī́ro 'a piece of news' → obl. *hīrós-/hīrón-*

Masculine nouns can be divided into three groups according to the oblique form: in the first group, only the oblique in *-es-/-en-* is used, in the second group, only the the oblique in *-os-/-on-*, and there is a third group where the two possible forms vary. The two competing patterns can be seen here next to each other throughout the whole paradigm in Table 1.

Table 1: The two masculine paradigms

Masculine	bakró 'sheep'		sókro 'father-in-law'	
	Singular	Plural	Singular	Plural
Nominative	bakró	bakré	sókro	sokrurá
Accusative	bakrés	bakrén	sokrós	sokrón
Dative	bakréske	bakrénge	sokróske	sokrónge
Locative	bakréste	bakrénde	sokróste	sokrónde
Ablative	bakréstar	bakréndar	sokróstar	sokróndar
Instrumental	bakrésa	bakrénca	sokrósa	sokrónca
Genitive	bakrésk-	bakréng-	sokrósk-	sokróng-
Vocative	bakrá	bakrále	sokrá	sokrále

We can draw up the following schema, shown in Figure 6, for the masculine oblique base, where N is a masculine noun. It contains the oblique marker *-es-/-en-* as the phonological form on the one hand, and the oblique marker *-os-/-on-* on the other.

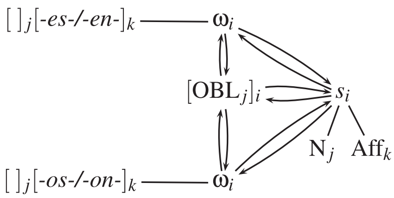


Figure 6: Schema for the masculine oblique base[8]

In this combination of two separate schemata, one containing the phonological form $\omega_i[\ ]_j[\text{es/en}]_k$ and the other one containing the phonological

[8] Although the oblique case is ultimately a morphological category, it can be split up into a syntactic and a semantic component. The syntactic component s_i covers the syntactic position and structure, while [OBL] denotes the semantic component in the schema. In Conceptual Semantics (cf. e.g., Jackendoff 2006), plurality, for example, is encoded as a function in the semantic structure. The oblique can also be considered as a function whose argument is the set of items being enabled to take further semantic functions on; the output of the function is the aggregate of such items.

form $\omega_i[\ ]_j[\text{os/on}]_k$, the same semantic content corresponds to two different phonological forms. The correspondence between the phonological form $\omega_i[\ ]_j[\text{es/en}]_k$ and the semantic content OBL_j is weakened by the presence of the other schema, where the same semantic content corresponds to a different phonological form, $\omega_i[\ ]_j[\text{os/on}]_k$, and this is also true the other way round: the correspondences between each phonological form and the semantic content OBL_j are weakened by each other.

To illustrate this, the masculine nouns we have from the newly collected data are listed in Tables 2–4. Only items which have at least one attested oblique form were taken into consideration. The tables contain 28 masculine nouns whose oblique form is *-es-/-en-*, 23 masculine nouns whose oblique form is *-os-/-on-*, and, in addition, there are nine lexical items whose oblique forms vary. In the tables, the words are grouped together in the order of the number of syllables (nouns with one syllable only appear among the ones with the oblique form *-es-/-en-*, while nouns with four syllables only appear among the ones with the oblique form *-os-/-on-*). Within the groups, the words are listed according to the end of the stem: whether there is a consonant, an /i/ or an /o/.

Table 2: Masculine nouns with the oblique form *-es-/-en-*

Noun	Attested oblique forms
	One syllable
berš 'year'	beršésko
del 'god'	devléske/dēvléske
gad 'shirt, clothes'	gādénca/gadéske/gādéske/gādénge/gādéngo
gav 'village'	gavéske
grast 'horse'	grastéske/grastén
kašt 'tree'	kaštéske/kaštésa/kašténge/kašténca
kher 'house'	kheréske/kherésko
kraj 'king'	krajéske/krajénge
murš 'man'	muršéske
nāj 'finger'	nājénca
rom 'Romani man'	roméske/roménca/romén/romés
than 'place'	thanéste/thanés
vast 'hand'	vastésa

Two syllables	
abáv 'wedding'	abavéske
bijáv 'wedding'	bijavéske
gurúv 'bull'	guruvén
kotór 'cloth'	kotorésa
manúš 'man'	manušés/manušén/manušéste/manušésko/ manušéstar/manušénca/manušénge
bāló 'pig'	bālén
gāžó 'non-Romani man'	gāžéske/gāžéstar/gāžén
kurkó 'week'	kurkéstar
šāvó 'boy'	šāvéske/šāvés/šāvén/šāvénge/šāvénca
Three syllables	
gḗzeši 'train'	gēzešésa
koldúši 'beggar'	koldušéstar/koldušés/koldušén/koldušénca
kopā́či 'tree trunk'	kopāčéske
pohā́ri 'glass'	pohārénca

Table 3: Masculine nouns with the oblique form *-os-/-on-*

Noun	Attested oblique forms
Two syllables	
ā́tko 'curse'	ātkónca
búso 'bus'	busósa
čā́so 'hour, watch'	čāsóngo
fṓro 'town'	fōróske
gíndo 'problem'	gindóstar/gindónca
hī́ro 'a piece of news'	hīróstar
nā́so 'child's father-in-law'	nāsósko
nī́po 'relatives'	nīpósa/nīpós
pújo 'chicken'	pujón
ríto 'field'	ritóske
trájo 'life'	trajósko

	Three syllables
āláto ‘animal’	ālatón/ālatós
bārṓvo ‘baron’	bārōvóske
čalā́do ‘family’	čalādós/čalādósa/čaládón
falató ‘a little bit of food’	falatóske/falatón
xāmásko ‘food’	xāmaskós
jōsā́go ‘livestock’	jōsāgós
laptópo ‘laptop’	laptopósa
sómsḗdo/sómsī́do ‘neighbour’	somsēdósko/somsēdóski/somsīdós/somsēdós/somsēdóske
vonáto ‘train’	vonatósa
	Four or more syllables
ternimā́ta ‘the young ones’	ternimātós/ternimātóske/ternimātónca/ternimātónge
šegīččḗgo ‘help’	šegīččēgóske/šegīččēgós
sā́mītōgḗpo ‘computer’	sāmītōgēpósa

Table 4: Masculine nouns where there is variation

Noun	Attested oblique forms
	Two syllables
sókro ‘father-in-law’	sokróske/sokrónge/sokrénge
	Three syllables
bašadó ‘telephone, mobile’	bašadésa/bašadósa
čókano ‘hammer’	čokanésko/čokanósko
dúhano ‘tobacco’	duhanés/duhanéski/duhanós/duhanóski
kirā́ji ‘king’	kirājéske/kirājénge/kirājén/kirājón
mobiló ‘mobile phone’	mobilésa/mobilósa
pokrṓco ‘blanket’	pokrōcésa/pokrōcósa
	Four syllables
kirčimā́ri ‘bartender’	kirčimārésa/kirčimārósa/kirčimāréstar/kirčimāróstar/ kirčimārénca
telefóni/telefóno ‘telephone’	telefonésa/telefonósa

As for the stems whose oblique forms vary, the variation is slight in some cases, with one or the other more dominant, but there are cases, like *dúhano*, where we find that the amounts of the two different oblique occurrences are basically equal.[9] The overall proportion of the frequency of the stems with the oblique forms *-es-/-en-, -os-/-on-* and the stems where the forms vary can be seen in Table 5.

Table 5: Number and proportion of the frequency of the stems with the oblique forms *-es-/-en-, -os-/-on-* and the varying stems

Oblique form	Number	Percentage
-es-/-en-	28	47%
alternating	9	15%
-os-/-on-	23	38%

The varying stems and the total number of occurrences of both variants in the data are repeated in Figure 7 (overleaf), except for two items, where the variation is very slight and needs further evidence: there is only one instance containing the suffix *-en-* for *sókro* 'father-in-law' and there is only one instance containing the suffix *-on-* for *kirā́ji* 'king'.

Variation seems to appear more often among words where the final vowel of the nominative singular form is /o/, see for example *čokáno* 'hammer', *dúhano* 'tobacco', *mobī́lo* 'mobile phone', *pokrṓco* 'blanket', *telefóno* 'telephone'.

The word *telefóno* has apparently got an alternative nominative form, *telefóni*, and there are some other masculine nouns ending in /i/ which show variation, like *kirčimā́ri* 'bartender', *kirā́ji* 'king'. The fact that we may find variation in the oblique form of lexical items the nominative singular ending of which is *-i* needs further investigation and confirmation when we have more data at hand. The fact that the oblique form of the word *telefóni/telefóno* 'telephone', for example, appears both as *telefonés-* and as *telefonós-* might as well be the result of the different nominative forms. Similar instances have been attested, for example the coexistence *mǖšoró* and *mǖšorí* 'programme'. With regard to the variation in the words *kirčimā́ri* 'bartender' and *kirā́ji* 'king' we must note that there are ambiguous cases, but there is not enough information avail-

9 More evidence for the variation comes from Cech et al. (1999), who provide a further example: the oblique form of the word *kókalo* 'bone' appears as both *kokalós-* and *kokalés-*.

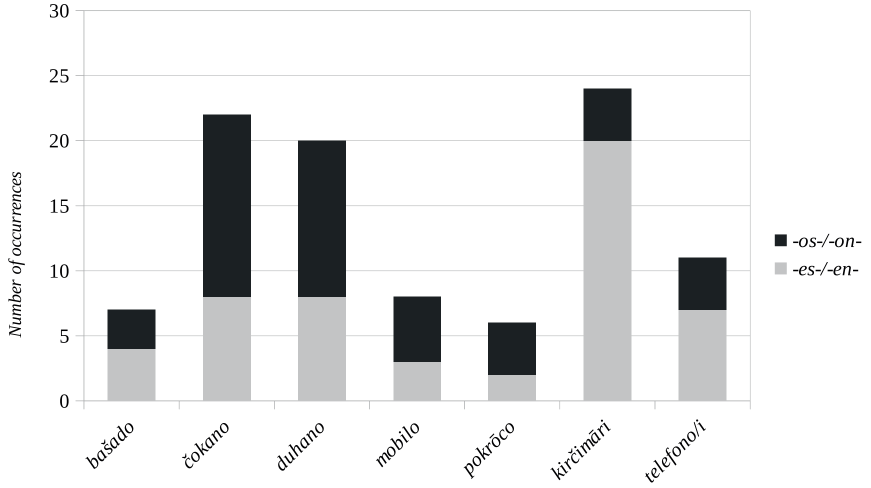


Figure 7: The total number of occurrences of the varying masculine stems in the data

able to draw a conclusion from them. While in the Kalderaš dialect, Boretzky (1994) documents oblique forms with *-es-/-en-* only for nouns with a stem-final /i/, for example *limóri* 'grave' ~ *limorés-/limorén-*, Cech and Heinschink (1999) only quote masculine nouns with a stem-final /i/ where the oblique suffixes are *-os-/-on-*, for instance *juhā́si* 'shepherd' ~ *juhāsós-, doktóri* 'doctor' *doktorós-* etc. The newly collected Northern Vlax Romani data from Hungary show that the situation is not so straightforward.

4.2. Possible causes and explanations

4.2.1. Variation in the position of stress

In this section, we will look at the relationship between the variation in the position of stress and the appearance of one or the other oblique form and we will see that even though one is not the direct consequence of the other (as the choice of words in Table 1, where the two different patterns are presented, intentionally suggests), there is certainly correlation between the two aspects, which means that there are certain other factors that we might want to take into consideration besides the stem-final vowel.

A possible cause of the variation in the oblique forms, which needs further investigation, is the variation in stress. Generally, and especially for disyllabic words, where the stress falls on the last syllable of the nominative singular form, there is no variation, the oblique suffix will be *-es-/-en-*, and where the stress falls on the first (penultimate) syllable, the oblique suffix will be *-os-/-on-*. No matter what the oblique ending is and where the stress falls in the nominative singular form, the stress in the oblique forms always falls on the oblique ending, so *bakró* 'sheep' will give *bakrés-*. On the level of the word, so on the surface, this results in penultimate stress: dative *bakréske*, locative *bakréste*, ablative *bakréstar* and instrumental *bakrésa*. A child who is acquiring Northern Vlax Romani as their mother tongue can base their assumptions concerning the oblique form on stress in case of disyllabic words.

For words with three syllables, stress is also lexical, but as there are more syllables, there are more options, and there is no such straightforward correlation as in the case of disyllabic words, where one position in the nominative predicts one oblique form and the other position predicts the other oblique form. While the oblique form of trisyllabic words will always have penultimate stress, the stress of the nominative form can fall anywhere between the first through the penultimate to the last syllable.

As we can see in Tables 2–4, the position of the stress cannot unambiguously predict the oblique form. While it is true that words with stem-final stress take the oblique forms *-es-/-en-* without exception, the oblique form of words where the stress shifts to a penultimate or ante-penultimate position is not so obvious. The words *padlṓvo* 'floor' and *rablṓvo* 'robber, highwayman', for example, have the oblique forms *padlōvés-* and *rablōvós-*, respectively (cf. Vekerdi 1985), in spite of the fact that both have penultimate stress. As we can see from the newly collected data, the oblique form of certain stems vary, for example *mobílo* 'mobile phone' ~ *mobilés-/mobilós-* or *dúhano* 'tobacco' ~ *duhanés-/duhanós-*. The choice of pattern may further be complicated by the fact that the stress of the nominative form may even vary within one stem, for example *kóčiši/kočíši* 'coachman' (Vekerdi 1985). In sum, where stress is not in a straightforward relationship with the oblique (that is, in words with three or more syllables), the vowel of the oblique suffix will become unpredictable and the oblique form may even begin to vary for individual items.

4.2.2. The number of syllables

There might be a correlation between the number of the syllables a noun has and the degree of variation it shows concerning the oblique forms. This is what we will examine in this section, eventually coming to the conclusion that the higher the number of syllables is, the more likely it is that the oblique form will vary.

Monosyllabic nouns always end in a consonant and invariably take the same oblique pattern, so *drom* 'road' and *dromés-* 'road' obl. This pattern is valid for other nouns ending in a consonant that are disyllabic, so *rašáj* 'priest' and *rašajés-* 'priest' obl. The other pattern appears when two factors, namely disyllabicity and a stem-final vowel present themselves simultaneously. A stem-final vowel introduces a certain amount of disturbance in the system, because it conflicts with the initial vowel of the oblique suffix, which is straightforward for consonant-final stems. Among disyllabic stems with a stem-final /o/, however, there is no variation in the strict sense: every lexical item which has two syllables and a stem-final /o/ will choose either one or the other pattern, and the position of the stress (final or penultimate) appears to be a reliable clue in this case, as seen in Tables 2–4: stress on the last syllable predicts the *-es-/-en-* form, while penultimate stress predicts the *-os-/-on-* form.

When the number of syllables rises to three, variation on the level of lexical items begins, and it is both intra- and inter-speaker variation. This means that the longer a word is, the more uncertain it gets which oblique stem it will take. As mentioned above, there is only slight variation for words longer than two syllables which end in a different vowel, like /i/: the frontness of the stem-final vowel will dominantly predict (or trigger) a front vowel in the oblique form. The back vowel /o/ of nouns with three syllables, however, will not be able to predict the oblique form unambiguously, just like disyllabic nouns ending in /o/ cannot.

Although there must be some among the trisyllabic masculine nouns with a stem-final /o/ that take *-es-/-en-* as their oblique (as attested in Vekerdi 1985, for example), our newly collected data do not contain them. However, they contain nine items with the oblique form *-os-/-on-* and four items whose oblique forms vary. This is somewhat in line with the varying stress pattern of trisyllabic nouns: the increase in the number of syllables increases the chance of variation, too. While the oblique form of disyllabic nouns never varies (it is either *-es-/-en-* or *-os-/-on-*), when the number of syllables exceeds two, the oblique form does begin to vary. This is further corroborated by the two items with four syllables: *kirčimā́ri* 'bartender' and *telefóni/telefóno* 'telephone'.

It should also be noted in connection with the higher number of *-os-/-on-* oblique forms that when variation begins, that is, at the level of trisyllabic nouns, the stem-final /o/ might tip the scales in favour of the oblique form which contains an /o/ (whereas the word *kirčimā́ri* 'bartender', with a stem-final /i/, seems to prefer the *-es-/-en-* forms).

4.3. Summary

In this section, we had a look at the first weak point in the morphology of Northern Vlax Romani, the masculine oblique base, in more detail. Following the description of the phenomenon in question, we went over two possible reasons for the weakness and the ensuing variation, and we found the following.

1. The position of stress. We saw that the stress pattern of disyllabic words (word-initial or word-final) corresponds to the choice of the oblique marker: word-initial stress corresponds to *-os-/-on-*, word-final stress corresponds to *-es-/-en-*. Stress begins to vary in trisyllabic words, and the same lexical item can occur with different stress patterns. That is exactly where the oblique markers begin to vary, too, so the varying stress pattern pairs up with the unpredictability of oblique forms.

2. The number of syllables. We found that while the oblique forms of disyllabic nouns do not vary, the oblique forms of trisyllabic nouns with a stem-final /o/ do. Based on this, it seems that the number of syllables influences the choice of oblique forms: the higher the number of syllables is, the higher the possibility of variation is.

5. A brief sidetrack: the "inherited–borrowed dichotomy"

We must mention here that in connection with the two different patterns, many (e.g., Boretzky 1989; Bakker 1997; Matras 2002) emphasise the existence of a strict morphological split between the vocabulary inherited from Indo-Aryan (as well as words borrowed from Persian and Armenian) and the vocabulary borrowed later from Greek and other (Romanian, Serbian, Hungarian etc.) contact languages.

> "The curious thing in Romani is that the newly arisen classes had not remained closed and limited to their constituting, i.e., Greek, lexical stratum. On the contrary, the athematic classes have become the only ones which exhibit any degree of contact productivity. Basically all post-Greek noun loans have been integrated into the new, athematic, rather than the old, thematic, classes."[10]
>
> (Elšík 2000, 17)

[10] In the English language literature focussing on Romani, the terms "thematic" or "oikoclitic" and "athematic" or "xenoclitic" are used to refer to the inherited and borrowed components, respectively.

Within the nominal inflection, this would mean that the *-es-/en-* pattern is used to inflect inherited nouns due to historical reasons, while the *-os-/-on-* pattern, being itself borrowed from Greek (Bakker 1997), is used to inflect borrowed nouns. For example, descriptions of Lovari (Hutterer & Mészáros 1967; Cech & Heinschink 1999) go along this path, with minor differences, so even masculine nouns with a stem-final *-i* take the oblique in *-os-/-on-* (Cech & Heinschink 1999, 22), which is clearly not the case, as we saw in section 6.3. Elšík (2000) discusses the historical development of nominal paradigms in detail, and, regarding the Greek-derived word *fṓro* 'town', he states that diachronically *fōrós-* replaced *fōrés-*, so that the oblique form could resemble the nominative singular. However, even in a diachronic sense, this is hard to justify, as it goes against the basic layout of the inherited inflection, where the oblique singular stem ends in *-es-*, no matter what the nominative ending is (for example nominative singular *bāló* 'pig' and obl. sing. *bālés-*).

Psycholinguistic factors might interfere in the form of the extent to which a native speaker "feels" that a certain word is borrowed or not, but this is very difficult to measure. Intuitively, one would think that, although the word *dúhano* is an earlier loan from Serbian than the word *čókano* from Romanian, the similarity of Hungarian *dohány* might evoke a sense of the word being less old. The fact that there is only slight hesitation concerning the oblique forms of *sókro*, a word borrowed from Romanian, does not really justify this as the current speakers of Northern Vlax Romani in Hungary have no access to Romanian at all. If the most important factor were the inherited or borrowed nature of a word, then, without direct access to the donor language, this factor would start to become obscure and there should be more hesitation, or, alternatively, the nominal classes would remain absolutely rigid, with no variation at all.

All in all, we have to dismiss the notion of the strict inherited-borrowed dichotomy, and thus, its erosion and any 'interaction' (Elšík 2000, 23) between the two layers, too. The two layers do not exist as there are no two specific and unique morphological systems used for one and the other; their inflection, strictly taken, is not different. What we must see clearly is that there are two patterns existing within the masculine paradigm of nouns ending in *-o*, and the choice may depend on several factors, including the overall frequency of the patterns. It is also true that the predominance of *-os-/-on-* forms in the case of *sókro,* for example, can be the result of the frequency of the forms of the particular paradigm itself (token frequency applied to paradigms), like in the case of the paradigm of *fṓro* 'town', where high token frequency may be the reason for the apparent lack of variation. On the other hand, variation in the case of the oblique form of a word like *čókano* 'hammer' can be the result of its lower token frequency. Other cognitive processes might play a role, too. For example, the extent of embeddedness is difficult to measure, but it may consist of such factors as how deeply embedded the word is mentally in language use, or what other notions might come into play, like even intuitions concerning the "Gypsyness" of the word.

6. The feminine oblique plural base

In this section, we will look at the second weak point, the feminine oblique plural base, in more detail. Following the description of the phenomenon, we will examine two possible aspects that might influence the choice of the plural oblique ending for feminine nouns. The two aspects are the following.

1. The masculine oblique plural *-en-*. Besides *-an-*, the other variant of the feminine oblique plural marker is *-en-*. The form is identical to one of the variants of the masculine oblique plural marker. As the semantic content (oblique plural) is also identical, we would like to look into the possible analogical influence of the masculine oblique plural marker on the feminine one. As we will see, the *-en-* form is dominant in both the masculine and the feminine nominal paradigms, which suggests that the mutual influence exists.

2. The feminine nominative plural suffixes. We will examine whether the nominative plural endings *-i* and *-a* have any connection to the appearance of one or the other plural oblique marker. We will find that there seems to be a relationship, which is made slightly more complicated by the fact that the singular ending of the nouns with the plural ending *-i* is *-a* and that of the nouns with the plural ending *-a* is often *-i*.

6.1. Description of the phenomenon

The feminine oblique singular base has one single form: *-a-*, so the oblique form of *šej* 'girl' is *šejá-*. However, the oblique plural base has got two possible forms: one is *-an-*, so the oblique plural base of a word like *khajnʲí* 'hen' is *khajnʲán-*, but there is another one, *-en-*, for instance the oblique form of *rā́ca* 'duck' is *rācén-*. They occur simultaneously as the feminine oblique plural base on several points of the feminine paradigm. This suggests that we are dealing with two competing patterns again.[11]

Table 6 shows the two different feminine paradigms. Note that the oblique singular forms of feminine nouns are completely unaffected by variation: the singular oblique marker is invariably -*a*-.

Table 6: The two different patterns in the feminine

Feminine	rā́ca 'duck'		māčí 'fly'	
	Singular	Plural	Singular	Plural
Nominative	rā́ca	rācí	māčí	māčá
Accusative	rācá	rācén	māčá	māčán
Dative	rācáke	rācénge	māčáke	māčánge
Locative	rācáte	rācénde	māčáte	māčánde
Ablative	rācátar	rācéndar	māčátar	māčándar
Instrumental	rācása	rācénca	māčása	māčánca
Genitive	rācák-	rācéng-	māčák-	māčáng-
Vocative	rā́ca	rācále	mā́ča	māčále

The two different patterns can be represented by the following combination of two schemata, shown in Figure 8, where N is a feminine noun. The correspondence between the phonological form $\omega_i[\ ]_j[\text{an}]_k$ and the semantic content OBL PLUR$_j$ is weakened by the presence of the other schema, where the same semantic content corresponds to a different phonological form, $\omega_i[\ ]_j[\text{en}]_k$. We can also look at it from the other direction: the correspondence between the phonological form $\omega_i[\ ]_j[\text{en}]_k$ and the semantic content OBL PLUR$_j$ is weakened by the presence of the other schema,

[11] According to the literature (Matras 2002, 83; Elšík 2000, 22; Boretzky 1994, 33), the form -*an*- is the result of a renewal or assimilation on the basis of the oblique singular; in other words, it would be a differentiation process aiming at paradigmatic opposition. For example, the oblique plural base of a word like *krangá* 'branch' is supposed to be *krangán*- (Hutterer & Mészáros 1967, 49), from an original oblique plural in -*en*-, and this most often happens in the Vlax dialects. However, the plural oblique of *krangá* 'branch' exclusively appears as *krangén*- in the newly collected data. We could begin to speculate whether one or the other is the "original" form and whether, if *krangén*- was the original form, it could have been retrieved after an intermediary stage; whatever the case is, all this could suggest that the variation we see here might be a sign of an ongoing change.

where the same semantic content corresponds to a different phonological form, $\omega_i[\ ]_j[\text{an}]_k$.

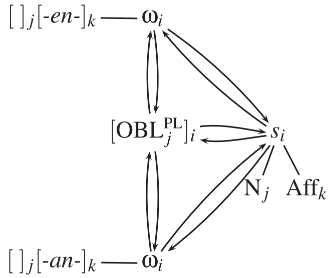


Figure 8: The combination of two schemata for the feminine oblique plural

The feminine nouns from the newly collected data can be seen in Table 7. The items are grouped together according to their oblique plural form; items with no attested plural oblique form were excluded. Out of the total twenty items, there are four whose oblique plural marker is *-an-*, there are seven items whose oblique plural marker is *-en-*, and there are nine stems where the oblique forms vary. A striking fact here is that the number of stems where there is variation is much higher than expected based on earlier sources, like Vekerdi (1985).

Table 7: Feminine nouns and their oblique forms from the newly collected data

Noun	Attested oblique forms	Noun	Attested oblique forms
Nouns with the oblique plural -an-		*Nouns with variation*	
xajíng 'well'	xajingánge/xajingángo	katt 'scissors'	kattʲánca/kattʲénca
khajnʲí 'hen'	khajnʲán	māj 'meadow'	māján/mājánge/mājénge
papín 'goose'	papinʲán/papinʲén	patrí 'leaf'	patrénca/patránca
pīrí 'saucepan'	pīránge	šūrí 'knife'	šūránca/šūrénca
māčí 'fly'	māčánca	tʲīrí 'ant'	tʲīránca/tʲīrénca
Nouns with the oblique plural -en-		bāj 'sleeve'	bājánca/bājénca
angrustí 'ring'	angrusténdar	bār 'garden'	bāránge/bārán/bārénge
armajá 'curse'	armajénca	bórotva 'razor'	borotvénca/borotvánca
cincā́ri 'mosquito'	cincārénca		
kangrí/krangí 'branch'	kangrénca/krangénca		
kúrva 'whore'	kurvéngo		
mesají 'table'	mesajéndar		
rā́ca 'duck'	rācén		

The overall proportion of the frequency of the stems belonging to the two feminine oblique plural patterns and the stems where the oblique forms vary can be seen in Table 8. We can see that the feminine class of nouns is even more affected by variation than the masculine class, with a higher percentage of all the attested stems showing variation.

Table 8: Number and proportion of the stems belonging to the two feminine oblique plural patterns and the varying stems

Oblique form	Number	Percentage
-*en*-	7	35%
alternating	9	45%
-*an*-	4	20%

The varying stems and the total number of occurrences of both variants in the data are repeated in Figure 9, except for two items, where the variation is very slight and needs further evidence: there is only one instance containing the suffix -*an*- for *bāj* 'sleeve' and there is only one instance containing the suffix -*en*- for *māj* 'meadow'.

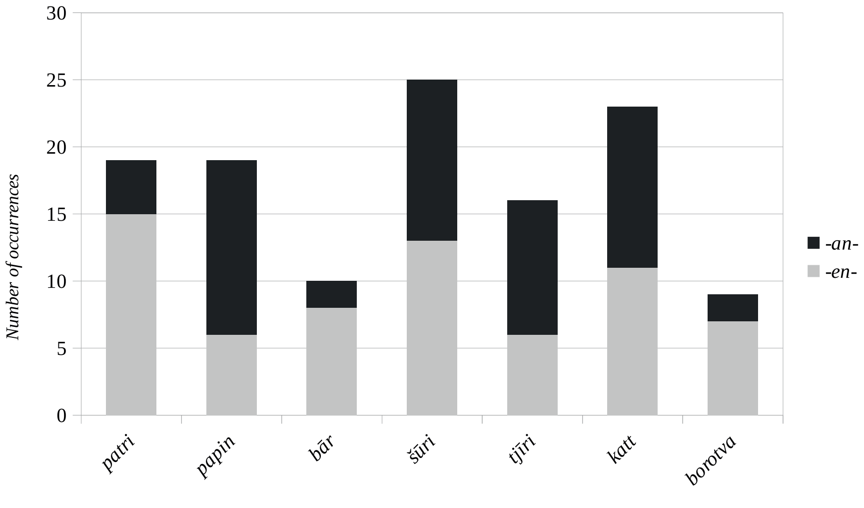


Figure 9: The total number of occurrences of the varying feminine stems in the data

We have to note here that Cech and Heinschink (1999) try to explain this again with the difference between inherited and borrowed words: -*an*- is used with inherited words and *-en-* is used for borrowed words. This is, however, completely inconsistent with the data and even with the way the inherited-borrowed dichotomy in the masculine is traditionally analysed, and thus should be dismissed.

The general frequency of /a/ and /e/ in the Romani verbal and nominal suffixes can play a role in the presence and competition of the two patterns, although this is contradicted by the fact that the proportion of the two different forms varies among the different stems. As we could already see, while the vowels /u/ and /i/ appear less often in suffixes in general, and even then they are more typically used in derivation, /e/ and /a/ are quite common in the inflection of Romani, for example as the vowel component of nominal oblique markers, both feminine and masculine, and of personal concord markers on verbs.

As we can see in Table 9, the personal concord markers for consonantal verbs (with the inclusion of the /e/ which was analysed as epenthetic by Baló 2008) exclusively contain these two vowels.

Table 9: Verbal personal concord markers

	1st sing.	2nd sing.	3rd sing.	1st plural	2nd plural	3rd plural
Present	-av	-es	-el	-as	-en	-en
Past	-em	-an	-as	-am	-an	-e

If we consider the fact that the first and second person plural forms are less frequent generally, we see that the proportion of personal concord markers containing /e/ and /a/ is 5 : 3, which corresponds to the tendencies we find for the distribution of the two vowels in the feminine oblique plural marker. Even if both the verbal and the oblique markers reflect a more general distribution or proportion of the vowels within the language, it is important to see that the distribution does not only present itself as different nominal classes formed with one or the other vowel, but also as stem-level variation, where one single stem can form the oblique with both markers.

The nominal oblique markers, including feminine nouns, can be *-es-, -en, -a-, -an-*, all containing /e/ or /a/. In addition, /o/ also appears in the variant oblique masculine forms *-os-* and *-on-*. The vowel /o/ is, however, not present elsewhere in the inflection. Considering all this, it follows that the variation in the feminine oblique plural between *-en-* and *-an-* is much more salient, with variation seen in nine stems out of 20, than the variation in the masculine oblique between *-es-/-en-* and *-os-/-on*, where there is variation in only nine out of 60 stems.

It is also important to note that the variation always includes /e/ as one of the elements of varying pairs of vowels: in case of the masculine oblique, the variation is /e/∼/o/, whereas in the feminine oblique plural it is /e/∼/a/. Its presence is in line with the overall high frequency of /e/, while the fact that it frequently takes part in some kind of variation is in line with the hypothesis that /e/ could be a default vowel and thus it is less stable. Let us not forget that it is always deleted where there is a thematic vowel at the end of the stem of the verb (cf. Baló 2008).

6.2. Possible causes and explanations

6.2.1. The masculine oblique plural *-en-*

The presence of the *-en-* pattern in the feminine may be connected to its simultaneous presence in the masculine. While the *-en-* pattern exerts a neutralising effect, making all plural paradigms look identical and decreasing the extent of gender difference, the *-an-* pattern exerts an opposite effect, trying to maintain an intra-gender uniformity, being more similar to the singular oblique marker *-a-*. A possible, additional aspect of variation is the presence of /n/ in the plural oblique across the whole nominal morphology; /n/ is a common trait of both the masculine and the feminine paradigms, so variation emerges more easily.

The correlation between the masculine oblique plural *-en-* and the feminine oblique plural *-en-* is shown in Figure 10, where the schemata for the masculine oblique plural and the feminine oblique plural are connected through a thick dashed bidirectional arrow, indicating mutual influence. However, as we will see in Figure 11, separating the masculine and the feminine phonological components containing the *-en-* suffix is not necessary at all; just like the syntactic component, the identical phonological components can be conflated into a single one as well.

Let us have a look at the phenomenon through the examples of *rakló* ‘boy’ and *rakljí* ‘girl’, which are apparently close cognates of each other, related to Sanskrit *laḍikka* ‘child’, with the feminine form derived through gender assignment.

As we can see from the example in Table 10, the forms show great uniformity, while maintaining opposition and differentiation. The back vowel of the nominative singular *rakló* is replaced by the front vowel /e/ in all other forms, while the front vowel of *rakljí* is replaced by the back vowel /a/ in the other forms. The opposition of the nominative singular end-

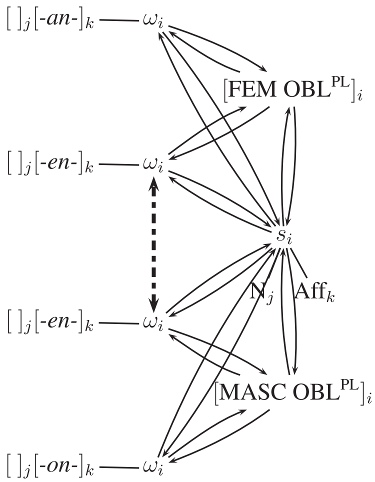


Figure 10: The relationship between the masculine and the feminine oblique plural endings

Table 10: Correlation between the masculine and feminine paradigms

Nominative singular	Nominative plural	Oblique singular	Oblique plural
rakló	raklé	raklés-	raklén-
rakljí	raklјá	raklјá-	raklján-

ings, /o/ and /i/, are swapped in the plural and in the oblique, but the front-back differentiation remains expressed. As we noted with regard to the masculine, disyllabic words always inflect the same way, having either /e/ or /o/ in the oblique ending. The word *rakló* belongs to the nouns which take *-es-/-en-*. The high degree of the similarity of the two words in the nominative singular maintains the contrast, but in case the word *rakljí* had forms like **rakljén-* in the plural oblique, so if there were variation, it would not really be surprising to see forms such as **raklón-* for the word *rakló*.

As stated before, the overall number of masculine nouns with the marker *-es-/-en-* is 28, as opposed to the 23 items with the marker *-os-/ -on-* (not counting the stems where there is variation). If we compare this to the seven feminine nouns with the oblique plural marker *-en-* and the four feminine nouns with the oblique plural marker *-an-*, we can see that, at least concerning type frequency, the *-en-* form dominates in both the masculine and the feminine paradigms, and the number of stems where there is variation is equal: nine in both paradigms. The fact that there are more feminine nouns which take the *-en-* form suggests that the dominance of the *-en-* form in the masculine influences the feminine paradigm indeed. The neutralisation effect is shown in Figure 11, where the masculine oblique plural and the feminine oblique plural converge in the ending *-en-*, and diverge through the endings *-an-* and *-on-*.

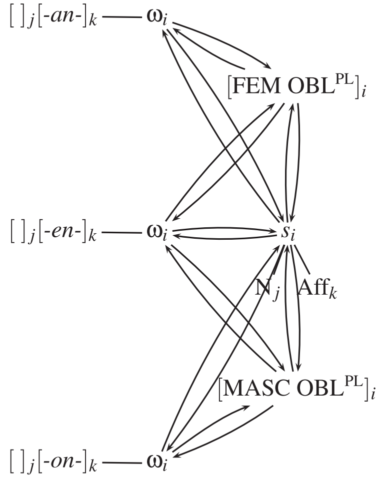


Figure 11: Combined schema of the masculine and feminine oblique plural

6.3. The feminine nominative plural suffixes

It would be appealing to say that the nature of the stem-final vowel plays a role in the choice of the oblique plural: if it is /i/, the vowel of the oblique plural marker is always /e/, if it is /a/, the vowel of the oblique plural marker is always /a/. However, as we could see from the data in Table 7, this is definitely not the case. On the other hand, there might be a possible and even more obvious connection between the nominative plural and the oblique plural. As we could see in Table 6, where the two patterns are introduced, the feminine plural form ends in /a/ if the nominative is /i/, so for example *pīrí* 'pot, saucepan'∼*pīrá* 'pots, saucepans', and it ends in /i/ if the nominative is /a/, see *kúrva* 'whore'∼*kurví* 'whores'. The oblique forms seem to correspond to the plural forms as for their backness.

(2) nominative singular *pīrí* → nominative plural *pīrá* → oblique plural *pīrán-*
nominative singular *kúrva* → nominative plural *kurví* → oblique plural *kurvén-*

If we have a closer look at the data, we find the following numbers and proportions. Out of the total 20 items, seven items follow the pattern. This means that if the nominative plural ending is /i/, they will take the oblique plural ending -*en*-, and if the nominative plural ending is /a/, they will take the oblique plural ending -*an*-, as seen in Table 11.

Table 11: Feminine nouns where the nominative plural ending corresponds to the oblique plural ending

Noun	Nominative plural form	Oblique plural form
	Nouns with the oblique form -an-	
xajíng 'well'	xajingá	xajingán-
khajn^{j}í 'hen'	khajn^{j}á	khajn^{j}án-
māčí 'fly'	māčá	māčán-
pīrí 'saucepan'	pīrá	pīrán-
	Nouns with the oblique form -en-	
armajá 'curse'	armají	armajén-
kúrva 'whore'	kurví	kurvén-
rā́ca 'duck'	rācí	rācén-

Four items behave in the opposite way, so their nominative plural ending is /a/ alongside the oblique plural ending -*en*-. There are no nouns whose nominative plural ending would be /i/ alongside the oblique plural ending -*an*-.

Table 12: Feminine nouns where the nominative plural ending does not correspond to the oblique plural ending

Noun	Nominative plural form	Oblique plural form
cincā́ri 'mosquito'	cincārá	cincārén-
mesají 'table'	mesajá	mesajén-
angrustí 'ring'	angrustá	angrustén-
kangrí/krangí 'branch'	kangrá/krangá	kangrén-/krangén-

The difference is significant, with almost twice as many items where there is correspondence in the backness.

Let us also check the tendencies among the seven stems where there is significant variation. Three of the stems where there is variation predominantly take either the nominative plural ending /a/ and the oblique plural ending -*an*-, or the nominative plural ending /i/ and the oblique plural ending -*en*-.

Table 13: Feminine nouns where there is variation with a bias towards the correspondence between the nominative plural and the oblique plural in backness

Word	Occurrences	pl. obl. -*en*-	pl. obl. -*an*-
papín 'goose'	19	32%	68%
t^jīrí 'ant'	16	37.5%	62.5%
bórotva 'razor'	9	78%	22%

On the other hand, two of the stems with varying forms go against the tendency, with the predominant pattern being that of the combination of the nominative plural ending /a/ and the oblique plural ending -*en*-.

Table 14: Feminine nouns where there is variation with a bias towards the opposition between the nominative plural and the oblique plural in backness

Word	Occurrences	pl. obl. -*en*-	pl. obl. -*an*-
patrí 'leaf'	19	79%	21%
bār 'garden'	10	80%	20%

Finally, there are two stems where the proportion of the two patterns is virtually equal, indicating a high degree of variation.

Table 15: Feminine nouns where there is a considerable degree of variation with no significant bias

Word	Occurrences	pl. obl. -*en*-	pl. obl. -*an*-
katt 'a pair of scissors'	23	48%	52%
šūrí 'knife'	25	52%	48%

In sum, we can say that the nominative plural ending can definitely or predominantly predict the corresponding oblique plural for eleven stems, while this prediction goes awry in case of only seven stems. This suggests that there is a tendency for the feminine nominal plural suffix to influence the choice of the oblique plural suffix, but it might be weakened by the fact that the nominative singular suffix is exactly the other way round. This is shown in Figure 12, where the schemata for the nominative plural and the oblique plural are connected through dashed arrows. The thick arrows represent the dominant direction of prediction, while the thin arrows show a weak correlation.

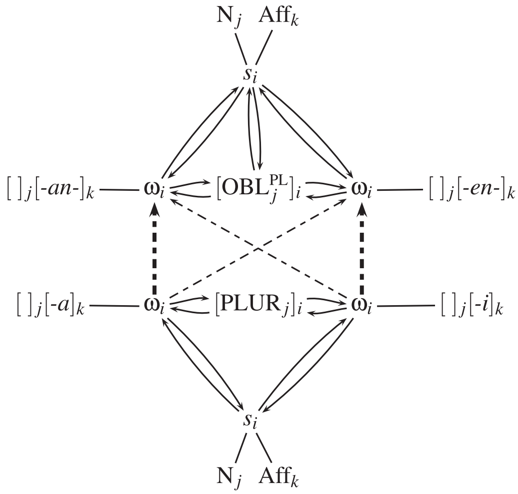


Figure 12: The relationship between the feminine nominative plural and the feminine oblique plural as shown in the form of schemata

6.4. Summary

In this section, we looked at the second weak point, the feminine oblique plural base, in more detail. Following the description of the phenomenon, we examined two possible aspects that might influence the choice of the plural oblique ending for feminine nouns, and we found that the two aspects seem to exert influence indeed.

1. The masculine oblique plural *-en-*. Besides *-an-*, the other variant of the feminine oblique plural marker is *-en-*, which is identical to one of the variants of the masculine oblique plural marker. We looked into the possible analogical influence of the masculine oblique plural marker on the feminine one. As we saw, the form *-en-* is indeed dominant in both the

masculine and the feminine nominal paradigms, which suggests that the influence exists.

2. The feminine nominative plural suffixes. We examined whether the nominative plural endings *-i* and *-a* have any connection to the appearance of the plural oblique marker *-en-* and *-an-*. We found that there is a relationship between the nominative and the oblique plural endings, with the front vowel /i/ predominantly predicting the marker *-en-* and the back vowel /a/ predominantly predicting the marker *-an-*. We also found an overall dominance of the marker *-en-*.

7. General conclusion

Through the example of the variation in the nominal morphology of the Northern Vlax Romani varieties spoken in Hungary, I would like to demonstrate that variation is an essential part of language and that its study brings us closer to a better understanding of the nature of language change, as language change is often preceded by variation. The study of variation, and especially intra-dialectal and intra-speaker microvariation might also provide us with some insights into the essential cognitive processes behind the structure and use of language. Although neurolinguistics is still in its infancy, based upon recent research in the field (Menn & Duffield 2014) it seems that construction-based and usage-based approaches can provide insights into how grammars can come closer to reflecting what our brains do. This "non-analytical" approach is also in line with recent experimental research in phonetics, speech perception and speech production (Port 2007; 2010). Apparently, in speech perception "the data strongly suggest that listeners employ a rich and detailed description of words" (Port 2007, 145) instead of abstract, segmented forms. In other words, "listeners encode particulars rather than generalities" (Pisoni 1997, 10).

I have also attempted to show that the simultaneous presence of two forces, regularisation on the one hand and differentiation on the other make language a dynamic process. For the study of variation and gradience, analogy proves a useful tool, especially because even variation can be gradient. This is illustrated through the phenomena we encounter in the nominal system of Northern Vlax Romani. Within the nominal morphology, we see two distinct, internally uniform patterns for both the masculine oblique forms and the feminine oblique plural forms. On the one hand, uniformity means that we do not find mixed paradigms; on the other, uniformity also refers to what we called regularisation above: the presence of the marker *-en-* in the feminine plural oblique is variation in the feminine plural paradigms but uniformity in the wider category of nouns.

Acknowledgements

I would like to thank the anonymous reviewer for their invaluable comments on the paper. The support of the Hungarian National Research, Development and Innovation Office (grants K 111961 and K 125596, principal investigator: László Kálmán) is gratefully acknowledged.

References

Adamou, E. 2016. A corpus-driven approach to language contact. Berlin & New York: Mouton de Gruyter.

Bakker, P. 1997. Athematic morphology in Romani: The borrowing of a borrowing pattern. In Y. Matras, P. Bakker and H. Kyuchukov (eds.) The typology and dialectology of Romani. Amsterdam & Philadelphia: John Benjamins. 1–21.

Bakker, P. 2001. Romani in Europe. In G. Extra and D. Gorter (eds.) The other languages of Europe. Clevedon: Multilingual Matters. 293–313.

Baló, M. A. 2008. A strange case of defectiveness in the lovari verbal paradigm. In L. Kálmán (ed.) Papers from the Mókus Conference. Budapest: Tinta Publishing House. 118–136.

Barbiers, S., L. Cornips and S. van der Kleij (eds.). 2007. Syntactic microvariation. Amsterdam: Meertens Institute.

Booij, G. 2010. Construction morphology. Oxford: Oxford University Press.

Boretzky, N. 1989. Zum Interferenzverhalten des Romani (Verbreitete und ungewöhnliche Phänomene). Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung 42. 357–374.

Boretzky, N. 1994. Romani. Grammatik des Kalderaš-Dialekts mit Texten und Glossar. Wiesbaden: Harrassowitz.

Boretzky, N. 2003. Die Vlach-Dialekte des Romani. Wiesbaden: Harrassowitz.

Brandner, E. 2012. Syntactic microvariation. Language and Linguistics Compass 6. 113–130.

Cech, P., C. Fennesz-Juhasz and M. F. Heinschink. 1999. Texte österreischer Lovara I. Arbeitsbericht 2 des Projekts Kodifizierung der Romanes-Variante der österreichischen Lovara (hrsgg. von Dieter W. Halwachs). Wien: Verein Romano Centro.

Cech, P. and M. F. Heinschink. 1999. Basisgrammatik. Arbeitsbericht 1a des Projekts Kodifizierung der Romanes-Variante der österreichischen Lovara (hrsgg. von Dieter W. Halwachs). Wien: Verein Romano Centro.

Elšík, V. 2000. Romani nominal paradigms: Their structure, diversity, and development. In V. Elšík and Y. Matras (eds.) Grammatical relations in Romani: The noun phrase. Amsterdam & Philadelphia: John Benjamins. 9–30.

Erdős, K. 1959. A Békés megyei cigányok. Cigánydialektusok Magyarországon [Gypsies in Békés county. Romani dialects in Hungary]. Gyula: Erkel Ferenc Múzeum.

Gaspar, L. 2012. Ciganos no Brasil. Pesquisa Escolar Online. Recife: Joaquim Nabuco Foundation.

Goldberg, A. E. 1995. Constructions: A construction grammar approach to argument structure. Chicago: The University of Chicago Press.

Hancock, I. F. 2013. The schooling of Romani Americans. An overview. In M. Miskovic (ed.) Roma education in Europe. Practices, policies and politics. New York: Routledge. 87–100.

Hutterer, M. and Gy. Mészáros. 1967. A lovári cigány dialektus leíró nyelvtana [A descriptive grammar of the Lovari Gypsy dialect]. Budapest: Magyar Nyelvtudományi Társaság.

Jackendoff, R. 2006. On conceptual semantics. Intercultural Pragmatics 3. 353–358.

Jackendoff, R. 2008. Construction after construction and its theoretical challenges. Language 84. 8–28.

Jackendoff, R. 2012. Language. In K. Frankish and W. M. Ramsey (eds.) The Cambridge handbook of cognitive science. Cambridge: Cambridge University Press. 171–193.

Kálmán, L., P. Rebrus and M. Törkenczy. 2012. Possible and impossible variation in Hungarian. In F. Kiefer, P. Siptár and M. Ladányi (eds.) Current issues in morphological theory: (Ir)Regularity, analogy and frequency. Amsterdam & Philadelphia: John Benjamins. 23–49.

Kozhanov, K. 2016. Is there a "new infinitive" in Russian Romani? A corpus-based study of subject–verb agreement in the subjunctive. Paper presented at the Conference on Corpus-Based Approaches to the Balkan Languages and Dialects, St. Petersburg, Russia.

Masica, C. P. 1991. The Indo-Aryan languages. Cambridge: Cambridge University Press.

Matras, Y. 2002. Romani: A linguistic introduction. Cambridge: Cambridge University Press.

Matras, Y. 2005. The status of Romani in Europe. Report submitted to the Council of Europe's Language Policy Division, October 2005.

Menn, L. and C. J. Duffield. 2014. Looking for a "gold standard" to measure language complexity: What psycholinguistics and neurolinguistics can (and cannot) offer to formal linguistics. In F. J. Newmeyer and L. B. Preston (eds.) Measuring grammatical complexity. Oxford: Oxford University Press. 281–302.

Miklosich, F. 1872–1880. Über die Mundarten und Wanderungen der Zigeuner Europas. Wien: Karl Gerold's Sohn.

Pisoni, D. B. 1997. Some thoughts on "normalization" in speech perception. In K. Johnson and J. W. Mullennix (eds.) Talker variability in speech perception. San Diego, CA: Academic Press. 9–32.

Port, R. F. 2007. How are words stored in memory? Beyond phones and phonemes. New Ideas in Psychology 25. 143–170.

Port, R. F. 2010. Rich memory and distributed phonology. Language Sciences 32. 43–55.

Rebrus, P. and M. Törkenczy. 2005. Uniformity and contrast in Hungarian. In L. Downing, T. A. Hall and R. Raffelsiefen (eds.) Paradigms in phonological theory. Oxford: Oxford University Press. 263–95.

Rebrus, P. and M. Törkenczy. 2011. Paradigmatic variation in Hungarian. In T. Laczkó and C. O. Ringen (eds.) Approaches to Hungarian 12: Papers from the 2009 Debrecen conference. Amsterdam & Philadelphia: John Benjamins. 135–162.

Rumelhart, D. E. 1980. Schemata: The building blocks of cognition. In R. J. Spiro, B. C. Bruce and W. F. Brewer (eds.) Theoretical issues in reading comprehension: Perspectives from cognitive psychology, linguistics, artificial intelligence, and education. Hillsdale, NJ: Lawrence Erlbaum. 33–58.

Tálos, E. 2001. A cigány és a beás nyelv Magyarországon [Hungarian Romani and Boyash]. In K. Kovalcsik, A. Csongor and Zs. Bódi (eds.) Tanulmányok a cigányság társadalmi helyzete és kultúrája köréből [Papers on the social and cultural aspects of the Romani]. Budapest: Eötvös Loránd Tudományegyetem – Iskolafejlesztési Alapítvány – Oktatási Minisztérium. 317–324.

Vekerdi, J. 1985. Cigány nyelvjárási népmesék [Gipsy dialectal folk tales]. Debrecen: Kossuth Lajos Tudományegyetem.

Vekerdi, J. 2000. A comparative dictionary of Gypsy dialects in Hungary. Budapest: Terebess.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 59–78

Recursive calculation abilities in agrammatic aphasia

A pilot study

ZOLTÁN BÁNRÉTI
Research Institute for Linguistics,
Hungarian Academy of Sciences
banreti.zoltan@nytud.mta.hu

ÉVA MÉSZÁROS
Eötvös Loránd University,
Bárczi Gusztáv Faculty of Special Education
meszaros.eva@barczi.elte.hu

KEYWORDS

recursion
arithmetic
language
aphasia

ABSTRACT

In a pilot study we found that agrammatic aphasia restricted the complexity of feasible arithmetical operations but left intact the ability of estimating quantities relative to one another as well as the ability to construct recursive sequences of figures and operations. Recursive numerical sequences and recursive operations were retained in the form of schemata or constructions. We argue for a common recursion module in the human mind that may be accessible for representations of arithmetical constructions, whereas the representations of linguistic constructions may be detached from that module in the case of Broca's aphasia.

1. Introduction

We conducted a pilot study in which we investigated possible impairments of recursive operations in arithmetical tasks. We started from the assumption that some prerequisites of arithmetical operations are sensitivity to structural relationships and the ability to perform recursive operations (cf. Hauser et al. 2002; Spelke & Tsivkin 2001; Krajcsi 2006). Therefore, we tested a healthy and a Broca's aphasic participant for their abilities to count and to carry out arithmetical operations. Beyond their comprehension of figures and the ability to estimate quantities, we were primarily interested in how much the participants retained of their sensitivity to structural features of arithmetical operations, especially the infinite recursion of sequences of figures.

2. Agrammatic aphasia and calculation

2.1. Varley et al. (2005) studied agrammatic aphasics' calculation abilities. Their motivation was that the grammars of natural languages and of arithmetical expressions exhibit some parallelism. These parallels include recursion and structure dependence. For instance, the computation of the correct result of numerical expressions involving subtraction or division: $(5 - 10; 10 - 5; 5 \div 10; 10 \div 5)$ or the ability to follow the bracketing of an expression $[5 \times (6 + 2)]$ requires awareness of the structural properties of the given expression. Similarly, recursive rule application allows for the derivation of a potentially infinite number of outputs from a finite set of constituents. This property is found both in natural language and the language of arithmetic (e.g., *The man that has a hat that has a brim that has a...*; $2 + 3 + 5 + 7 + \ldots + \ldots$). The interdependence of language and arithmetic can also be seen in devices like the "multiplication table", a way of encoding mathematical facts in a verbal form and storing the result in one's long-term memory. The content thus stored can be accessed with no computation load when it is needed in the solution of novel calculation tasks, minimizing the required overall computation load. The interaction between arithmetical procedures and the activation of learned verbal information leads to the hypothesis that the operation of multiplication can be especially sensitive to aphasics' linguistic limitations (Lemer et al. 2003).

In the case of unimpaired persons, during the execution of numerical tasks, a bilateral network of cerebral regions is activated to mirror operations of calculating the quantity of objects, sounds or other entities. Several studies have detected activity in the language centers of the left hemisphere when the task was to perform exact calculations with symbolic expressions (Cohen et al. 2000; Friederici et al. 2011; Friedrich & Friederici 2013). Among others, this was found in multiplication by one-digit numbers where the use of verbally encoded information is crucial: the frontal "linguistic" areas, including Broca's area, were found to be activated (Dehaene et al. 1999; van Harskamp & Cipolotti 2001; Delazer et al. 2003). On the other hand, in cases of aphasic language impairment, concomitant problems in calculation abilities have been attested (e.g., Cohen et al. 2000).

However, an alternative approach is also conceivable. Although arithmetical operations are carried out by processes that are also required for lexical and grammatical operations, by the time ontogenesis reaches adulthood, the architecture of the mature mind reserves a niche for counting that is independent of language. Some studies claim that in counting tasks, stronger activation shows up in the right hemisphere (in the intraparietal sulcus) than on the left side (Butterworth 1999; Dehaene et al. 2003). Some functional cerebral imaging techniques seem to suggest that "linguistic areas" are not active in calculation tasks (Pesenti et al. 2000; Zago et al. 2001). Some accounts claim that in developmental and acquired language impairments, linguistic and mathematical abilities may be dissociated, that is, they do not form a single system of abilities (e.g., Ansari et al. 2003). But such dissociations do not exclude the possibility that subsystems of the grammar and the lexicon do support calculation performance, even in the case of language impairment.

2.2. The studies by Varley et al. (2005) and Zimmerer & Varley (2010) were groundbreaking in that they focused on the issue of whether recursion and sensitivity to the peculiarities of hierarchical structure were parallel/interdependent properties of linguistic and arithmetical procedures.

Varley et al. (2005) studied three agrammatic aphasic persons. All three were university graduates, one of them had been a professor of mathematics until he was afflicted with aphasia. According to the test results, the calculation procedures of the three persons, including recursive operations and sensitivity to hierarchical structures, had remained intact, while they were moderate agrammatic aphasics in terms of both computer tomography (CT) results and performance in status tests. They performed relatively well on lexical comprehension and synonym finding. But they exhibited severe impairment of linguistic-syntactic abilities and produced guessing-level results in grammaticality decisions with respect to written sentences. They also showed asyntactic sentence comprehension and guessing-level results in understanding "reversible" sentences. Their spontaneous speech production consisted of broken phrases or constituents thereof. Varley and her colleagues administered meticulous subtests on the abilities of reading numbers as symbols and of identifying (mathematical) operators, these being prerequisites to performing well on calculation tests. Out of the three persons, only one was able to use lexical names of numbers in speech, while the other two were not. On the other hand, the calculation of quantities and their ratios turned out to be unimpaired for all of them.

The calculation tests were pen-and-paper-based and consisted of eight subtests: (i) estimating the relative positions of quantities along a vertical line; (ii) addition, subtraction, multiplication and division operations on integers, then (iii) addition and subtraction of fractions; (iv) multiplication both on the basis of the multiplication table and beyond it; (v) inverting an operation yielding a positive number into one yielding a negative number;

(vi) creating infinite sets of numbers; (vii) operations involving bracketing, where in some cases the brackets were syntactic in the sense that simply performing the operations left-to right would not give the correct result (e.g., $36 \div (3 \times 2)$), while in other cases the brackets were non-syntactic (e.g., $(3 \times 3) - 6$); and persons were also asked to (viii) generate bracketing (they received sequences of figures and operators with the instruction that they should insert brackets in several different manners and then calculate the results accordingly). We will return to the details of these subtests in our discussion of their Hungarian adaptations. In what follows, we will compare the performance of a healthy person and an aphasic participant. Varley et al.'s participants achieved good results in each of the subtests; in some cases they performed without a single error. The results show the mutual independence of structure-based linguistic vs. arithmetical operations within a given cognitive architecture. Although all persons were agrammatic aphasics, they applied syntactic principles in arithmetic appropriately.

Varley et al. (2005) and Zimmerer & Varley (2010) proposed two types of explanations of the interrelationship of the syntax of language and the syntax of arithmetic. According to one, the two systems work independently of each another, and the impairment of one does not need to concern the other. According to the other explanation, there is a shared syntactic system that underlies both language and arithmetic, but arithmetical processing may directly access this system without translating the expressions into a linguistic form first.

3. Participants

The aphasic participant was C, a 31-year-old right-handed man, 17 years of schooling, an engineer. He was assigned to aphasia type on the basis of CT results, the Western Aphasia Battery (WAB) tests (Kertesz 1982) and the Token test (De Renzi & Vignolo 1962). The WAB test and the Token test were adapted to Hungarian by Osmanné Sági (1991; 1994). The CT showed an isochemic stroke at the left arteria cerebri media. On the basis of the results of the Western Aphasia Battery (WAB), he was a Broca's aphasic with severe agrammatism, his Aphasia quotient (AQ) equalled 56 (in healthy participants: 93.8 or above; the maximum is 100). In the Token test he achieved 14 scores (healthy subjects above 32; the maximum is 36).[1] According to the CT results and the results in the WAB and Token tests, C exhibited the typical symptoms of severe agrammatic Broca's aphasia. C participated in our earlier investigation on the capacity of recursive sentence embedding. In those experiments C was not able to produce responses containing recursive sentence embedding, he gave only some simple, short, fragmented answers (Bánréti et al. 2016). He was severely impaired in producing recursive syntactic structures.

The healthy subject was Z, a 42-year-old right-handed man with 16 years of schooling, a teacher.

4. Materials and methods

To test our subjects' performance on arithmetic, we administered a variety of tasks based on Varley et al. (2005), a total of seven subtests. For details see the Appendix.

For the estimation task, they had to mark the approximate positions of 20 numbers (presented to them in a random order) along a 20 cm vertical line (number line) of which the two ends were marked as 0 and 100, respectively. The task probed into the degree of limitation of the subjects' utilization of quantity concepts. (The task sheets can be found in the Appendix.) The response was taken to be correct if the marking provided by the subject was within 5 mm from the proper value point. Next, addition (12 items), subtraction (12 items), multiplication (9 items), and division tasks (16 items) followed. The correct results were positive integers in all cases.

[1] The Western Aphasia Battery (WAB) uses a kind of standard protocol. Spontaneous speech is evaluated for articulation, fluency, content and presence of paraphasias. Comprehension is tested with yes or no questions, pointing commands, and one to three step commands. Naming is evaluated for objects, object parts, body parts, and colors. Repetition is requested for single words to complex sentences. The level of adequacy for reading and writing is also tested. Five subtests on fluency, information, comprehension, repetition, and naming impairment are classified from 0 to 10. The maximum result of each subtest is 10 points each. Accordingly, aphasia can be classified into global aphasia, Broca's aphasia, Wernicke's aphasia, transcortical motor, transcortical sensory conduction aphasia, and anomic aphasia types. For instance, in Broca's aphasia fluency ranges from 0 only to 4 points, comprehension ranges from 4 to 10, repetition is under 8 points and naming ranges from 0 only to 8. Aphasia quotient (AQ) shows the severity of aphasia. AQ is calculated by the addition of scores of the subtests and this sum is multiplied by two. The maximum is 100. Normal subjects score an AQ of 93.8 or above. An AQ around 50 shows a severe degree of aphasia, cf. John et al. (2017).

In the inversion task, two-digit numbers had to be subtracted or divided in a random order, such that first a smaller number had to be subtracted from a larger one (respectively, a larger number had to be divided by a smaller one) yielding a positive integer (e.g., $72 - 26; 60 \div 12$), then the other way round (yielding a negative number for subtraction and a fraction for division, e.g., $26 - 72; 12 \div 60$). All this was done in three instances.

In the bracketing resolution task, there were expressions involving syntactic bracketing (8 items) in which, if the subject followed just the linear order of operations without taking the brackets into consideration, the result would be incorrect, as in $36 \div (3 \times 2)$; and there were also expressions with non-syntactic bracketing (3 items) in which the correct result is obtained whether or not the brackets are taken into consideration, as in $12 \times (6 \times 7)$. Among the syntactic items, there were single and double pairs of brackets. In the latter case, another operation was embedded as a term of the main operation. While single bracketing occurred in the left term or in the right term double bracketing invariably occurred in the second term (8 items).

In the bracket generation task, the subject had to generate bracketing on sequences of four numbers linked by operators such that different ways of bracketing should yield different results, e.g., $(6 + 2) \times 5 + 8 = 48; 6 + (2 \times 5) + 8 = 24; 6 + 2 \times (5 + 8) = 32; (6 + 2) \times (5 + 8) = 104$; etc. The use of brackets in calculation tasks is taken to be an instruction for recursive operations as in these cases one or more terms of an expression are themselves results of a recursively embedded operation.[2]

In the infinity task, the subjects had to generate sequences of numbers. They had to find numbers larger than one but smaller than two, then after each response a number that is larger than the previous answer but still smaller than two, and so on – keeping on increasing the values without reaching the number two.

[2] Arithmetic operations have a default order: if only additions and subtractions are involved, their order does not matter. If a division or multiplication is one of the operations required and an addition or subtraction is the other, it is always the division/multiplication that comes first. Between division and multiplication, the order has to be signaled by bracketing, e.g., $(6 \div 3) \times 2 = 4$ but $6 \div (3 \times 2) = 1$. In complex expressions, it is the expression within the brackets that has to be calculated first; and within a pair of brackets, multiplication and division enjoy applicational precedence over addition and subtraction. For instance: $3 \times (20\text{–}\underline{5 \times 2}) = 3 \times (20\text{–}10) = 3 \times 10 = 30$. The default order (division/multiplication first) can be overridden by bracketing: $(6 + 2) \times 5 + 8 = 48; 6 + (2 \times 5) + 8 = 24; 6 + 2 \times (5 + 8) = 32$.

5. Results

5.1. Normal participant

The calculation tasks did not represent any difficulty for the normal participant except that he required a relatively long concentration of attention. The sporadically occurring errors may be due to that factor.

Tables 1 and 2 show percentages of errors in each task (n = all calculations performed by the subject, 0: percentage of erroneous calculations if no errors were made).

Table 1: Results of tasks in the arithmetical test in percentages of errors: normal participant

Subject	Estimation task $n = 20$	The four basic operations $n = 12$ /12 /10 /10				Subtraction and its inversion (yielding a negative number) $n = 6$	Division and its inversion (yielding a fraction) $n = 6$	Infinity $n = 11$
		+	−	×	÷			
Z	1.9	0	0	0	10	0	0	0

Table 2: Results of bracketing tasks in percentages of errors: normal participant

Subject	Bracketing operations			
	Single bracketing $n = 20$	Double bracketing $n = 8$	Generation of bracketing $n = 25$	Resolution of bracketing $n = 25$
Z	0	0	0	0

5.2. Agrammatic aphasic participant

Tables 3 and 4 show percentages of errors in each task (n = all calculations performed by the subject, 0: percentage of erroneous calculations if no errors were made).

Table 3: Results of tasks in the arithmetical test in percentages of errors: aphasic participant

Subject	Estimation task $n = 20$	The four basic operations $n = 12$ / 10 / 10 / 9				Subtraction and its inversion (yielding a negative number) n = 6	Division and its inversion (yielding a fraction) $n = 6$	Infinity $n = 10$
		+	−	×	÷			
C agrammatic aphasic subject	4.6	0	0	0	13	50	66.6	0

Table 4: Results of bracketing tasks in percentages of errors: aphasic participant

Subject	Bracketing operations			
	Single bracketing $n = 9$	Double bracketing $n = 3$	Generation of bracketing $n = 9$	Resolution of bracketing $n = 9$
C agrammatic aphasic subject	42.8	33.3	0	100

In the task involving the number line, C made few mistakes in localizing given numerical values along the line, the deviation amounted to 4.6%. (A 20 cm vertical line was at the subjects' disposal, so 2.5% difference meant 5 mm, C's average deviation – above 5 mm – only about 4 mm above the tolerance threshold). We can conclude that the notion of quantities represented by figures was unimpaired in both subjects.

With respect to the four basic operations, he was successful in addition and in subtraction. Here we found correct results for single-digit, two-digit, and three-digit numbers. In case of multiplication, only that of single-digit and two digit terms were done correctly, while no calculations with multiple digits were carried out at all. The division of a two-digit number by a one-digit number was correct, while C performed only one of the division tasks of three-digit numbers correctly; out of nine cases, he gave the wrong result in one case and gave a result that was roughly correct but not to the last decimal value in another.

Half of the inversion tasks yielded the wrong result in subtractions, and more than half of them in the case of divisions. In the case of negative results, C signaled by [–] that he would get a negative number, but most results were wrong. In the cases of dividing a smaller number by a larger one, he overlooked only in one out of three cases that the result would not be an integer.

The knowledge that sequences of numbers may be infinite was retained. He found numbers larger than one but smaller than two correctly; he gave several correct solutions and recognized the rule.

In bracketing operations, the order of operations in tasks involving a single pair of brackets was correct; the final results were not always correct due to calculation errors. In operations involving multiple brackets (embeddings), the partial calculations and the order of operations were correct, but the final result could not always be given.

In the last task, C was able to generate bracketing (embedding), he inserted brackets at different places in each case but failed to calculate the final results. In that task, he also used double bracketing.

6. Discussion

6.1. Impairments in linguistic resources

C had difficulties in verbalizing his calculations, but he was capable of self-monitoring. He showed several types of impairments in linguistic resources available for arithmetic operations, especially impairments in the lexical access of numerals. During the calculations, the digits were spontaneously read aloud, of which "9" was mistakenly read as "8" but in each case he corrected himself to "9". C hesitated typically at the verbal markers indicating place values; for example, he said: *nyolc... száz... őőő nem!... nyolc... VAN... hat* 'eight... hundred... hmm... no!... eight... TY... six'. In the end, he was always able to produce the correct name of the digit; he could encode the visual input into verbal form.

In his calculations, the names of the signs "+" and "−" were produced (called "plus" and "minus"). The verbal equivalents (names) of the division and multiplication were not used spontaneously. The operations $A \times B$ and $C \div D$ were called "*A* and *B*", "*C* and *D*". C did not use the words "division", "multiplication" either in nominal or in verbal functions (i.e., the phrases "*A* multiplied by *B*", and "*C* divided by *D*" were never realized). At the same time, the symbolic signs of multiplication and division were understood and the operations indicated by them could be performed, and their results were often accurate, even if not always. In other words, he performed the operations of multiplication, division, addition and subtraction without lexically accessing the exact names of those operations, except for using the expressions "plus" and "minus" for addition and subtraction, respectively.

C did not produce the names of the fractional numbers, he did not say "one third" or "two sixth", for digits like 1/3, 2/6, etc. but he called them *1 tört 3, 2 tört 6* '1 fraction 3, 2 fraction 6', etc. when he was asked to report on how he counted. At the same time, the results of addition and subtraction of the fractions were correct; he used the value of the common denominator of the fractions independently and correctly.

C did not use the term "bracket" either during the silent reading of the tasks or in the completion of the tasks containing brackets. He was able to produce the sequence and embedding of counting operations following the hierarchy required by the brackets.

6.2. Effects related to aphasic limitation

We can identify several different effects related to aphasic limitation. The first effect is that of complexity; in particular, the complexity of operations to be performed in each task. C was able to perform addition and subtraction without errors, while in division and multiplication three digit terms were avoided, inversion resulting in a negative number and the resolution of all bracketed formulae came with high error rates and the resolution of formulae generated by the subject himself proved to be even more difficult (100% error rate). Such robust effect of complexity is of course not surprising under the circumstances of severe agrammatic aphasia.

The second effect involves the participant's ability to estimate quantities within a given domain. C's error rate was only 4.6%. His ability in certain calculation procedures is more severely impaired than his ability to estimate quantities within a domain. C did not commit any errors with respect to the relative order of the quantities along the number line; the error rates werre due to cases in which they marked the points of the number line more than 5 mm off target. (C's average deviation – above 5 mm – was only about 4 mm). We can conclude that the notion of quantities represented by figures was unimpaired in the aphasic participant.

The third effect was the ability to generate sequences of numbers in a recursive manner. This was probed into by the task requiring the production of infinite sequences of numbers (Appendix VIII), and by the task asking for the generation of bracketing (Appendix VII) – including multiple bracketing – of the same series of numbers in several different ways. In those two tasks, C performed without error. He was able to produce sequences of numbers and sequences of operations that recursively contained other sequences of numbers and operations, respectively. However, C was not able to do the actual calculation on task (viii). He did not produce erroneous results but rather deemed the calculation of the value of the formula he had produced himself to be too difficult and gave up without trying.

In sum, the agrammatic aphasia did affect (limit) the complexity of calculations but left the ability intact to estimate relative distances of quantities and the ability to create recursive sequences of numbers or operations. The latter was done in terms of schemas/constructions, the calculations yielded concrete numerical end products were avoided.

7. Conclusion

The dissociations outlined above are interesting especially in view of the fact that in earlier studies we found strong limitations of linguistic-syntactic recursion in Broca's aphasia (Bánréti et al. 2016). These observations are now complemented by the finding that, in the case of arithmetical operations, albeit complexity effects show up similarly, the relative estimation of quantities and the ability of generating recursive sequences of numbers and operations can be retained in Broca's aphasia. The arithmetic operations by the aphasic person show limitations, but these do not concern the basic operations themselves, only their more complex versions. The difficulties in accessing the verbal linguistic resources that are useful for counting may lead to errors or confusion in more complex calculations but do not make them inaccessible. Recursive numerical sequences and recursive operations are retained in the form of schemata or constructions.

Some patterns of linguistic and arithmetic expressions have similarities, such as recursiveness and structure dependency. A moderate aphasic condition exhibits strong limitations in those linguistic (primarily syntactic and lexical) capacities, but arithmetical operations show a much better state preserving basic operations. This provides arguments for a model in which linguistic and arithmetic processes are separated, and counting may be kept separate from language in adult age. In this model recursive arithmetical operations can be carried out inspite of linguistic impairments.

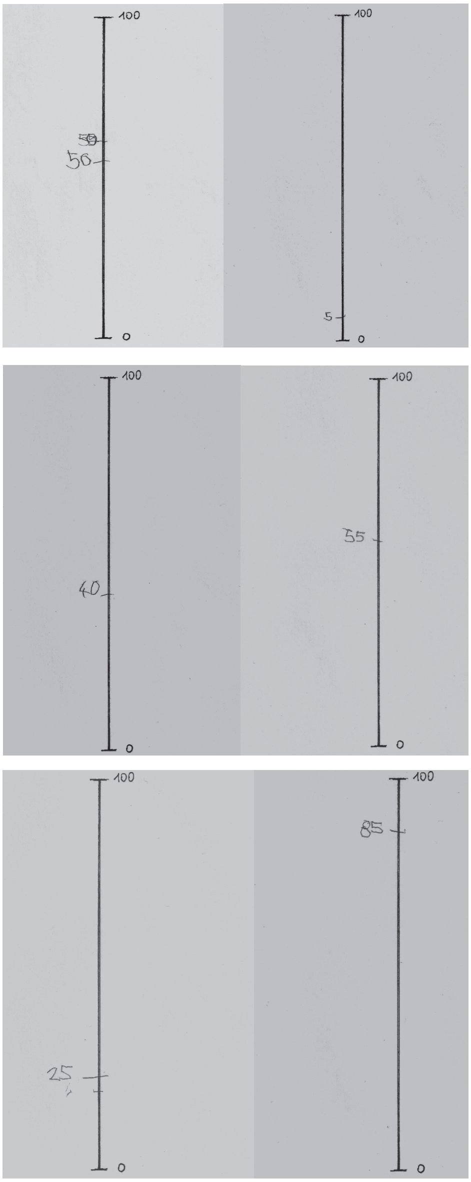
Our results support the model proposed by Zimmerer and Varley (2010) that posits a module of recursive operations in the human mind that are shared (among others) by linguistic and arithmetical performance. This common recursion module may be accessible for representations of arithmetical constructions, whereas the representations of linguistic constructions may be detached from it in the case of Broca's aphasia. Varley et al. (2005) point out that in adult age[3] arithmetic can be sustained without the grammatical and lexical resources of the language.

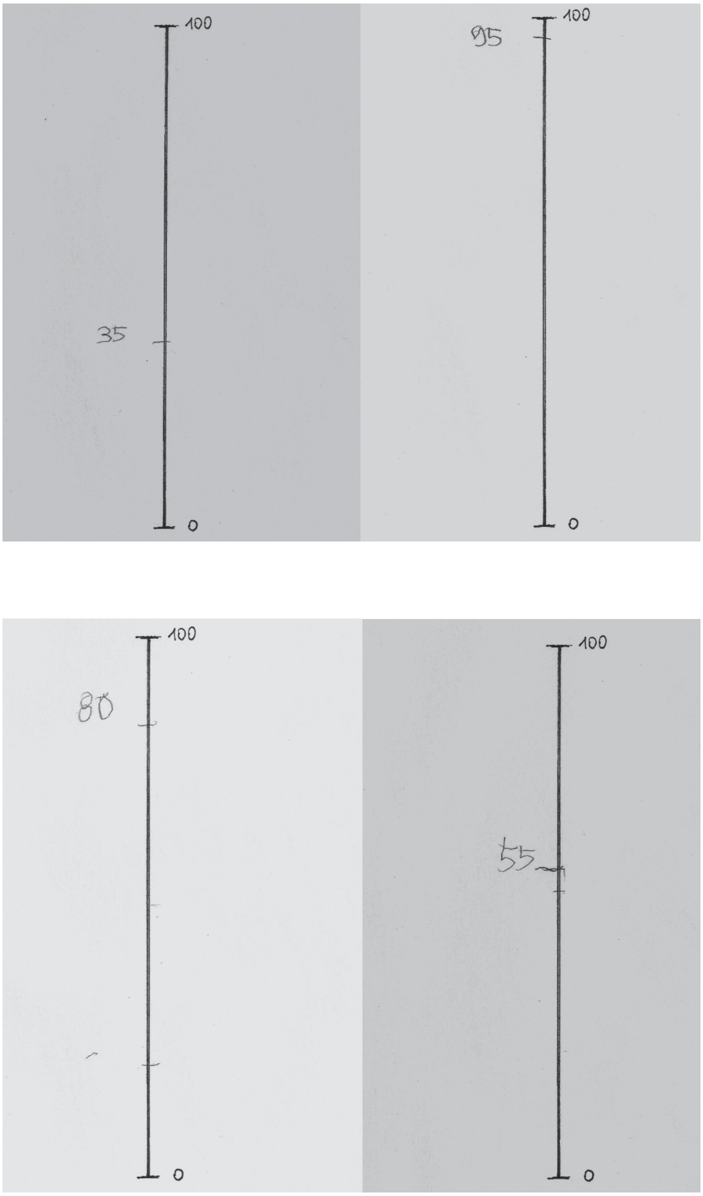
[3] Varley et al. (2005, 6) state: "Number words may be important in children's acquisition of numerical concepts and their digital, orthographic, phonological, and sensory representations. Similarly, language grammar might provide a 'bootstrapping' template to facilitate the use of other hierarchical and generative systems, such as mathematics. However, once these resources are in place, mathematics can be sustained without the grammatical and lexical resources of the language faculty. [...] grammar may thus be seen as a co-opted system that can support the expression of mathematical reasoning, but the possession of grammar neither guarantees nor jeopardizes successful performance on calculation problems".

Appendix

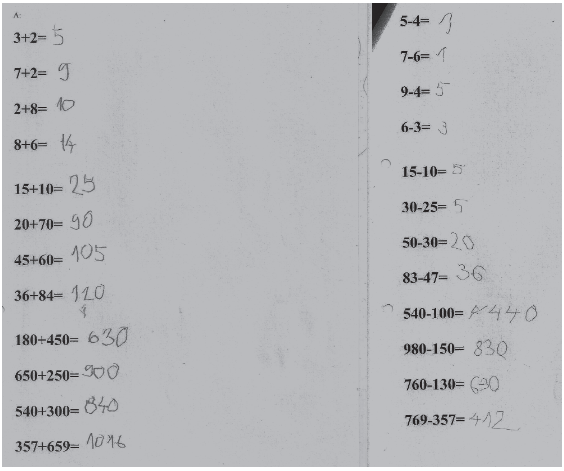
Aphasic participant: some examples

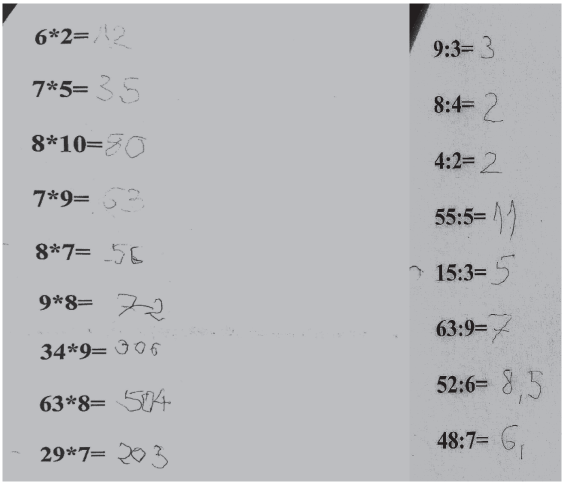
I. Estimation tasks

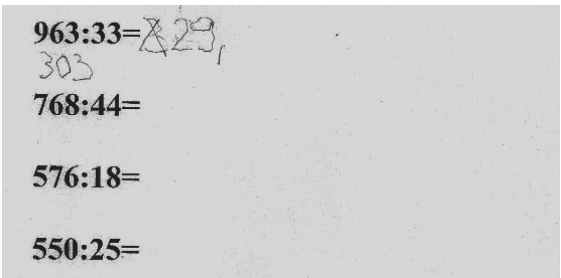



100
35
0
100
95
0
100
80
0
100
55
0

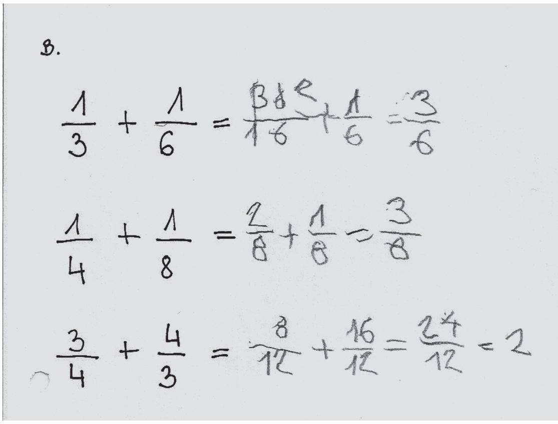
II. Basic operations: addition, subtraction, multiplication, division tasks


A:
3+2= 5
7+2= 9
2+8= 10
8+6= 14
15+10= 25
20+70= 90
45+60= 105
36+84= 110
180+450= 630
650+250= 900
540+300= 840
357+659= 1016
5-4= 1
7-6= 1
9-4= 5
6-3= 3
15-10= 5
30-25= 5
50-30= 20
83-47= 36
540-100= 440
980-150= 830
760-130= 630
769-357= 412

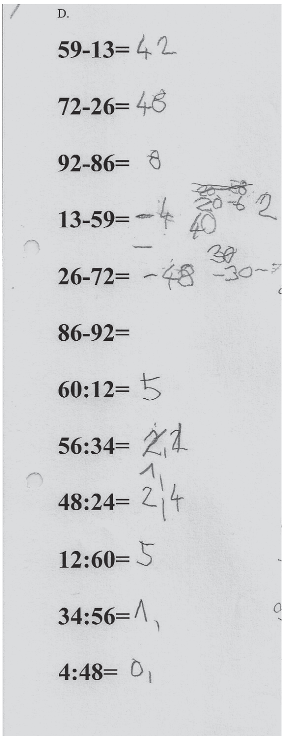

6*2= 12
7*5= 35
8*10= 80
7*9= 63
8*7= 56
9*8= 72
34*9= 306
63*8= 504
29*7= 203
9:3= 3
8:4= 2
4:2= 2
55:5= 11
15:3= 5
63:9= 7
52:6= 8,5
48:7= 6,


963:33= 329,
303
768:44=
576:18=
550:25=

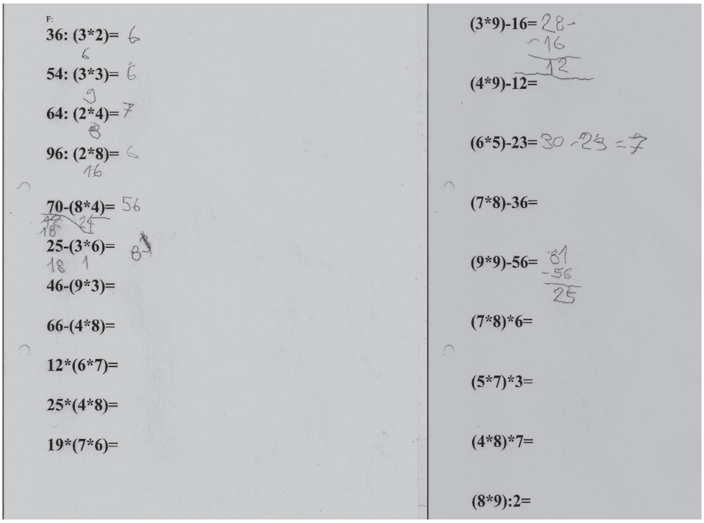
III. Addition of fractional numbers, finding their common denominators



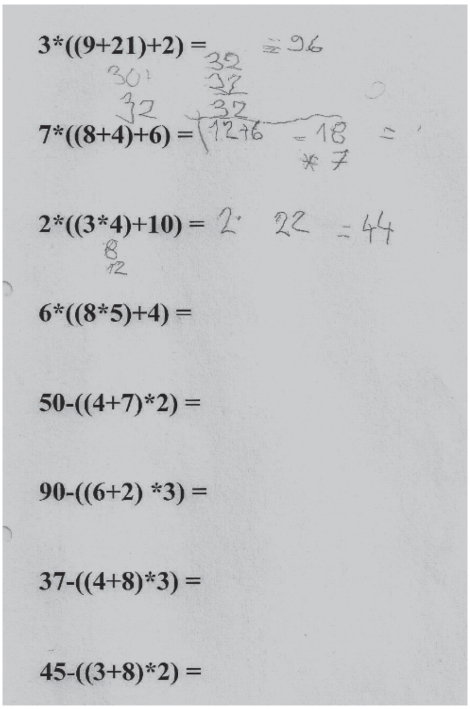
IV. “Inversion” tasks



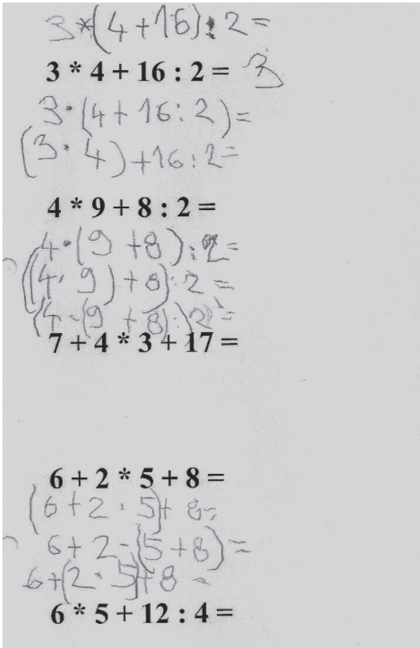
V. Single bracketing tasks


F:
36: (3*2)=
54: (3*3)=
64: (2*4)=
96: (2*8)=
70-(8*4)=
25-(3*6)=
46-(9*3)=
66-(4*8)=
12*(6*7)=
25*(4*8)=
19*(7*6)=
(3*9)-16=
(4*9)-12=
(6*5)-23=
(7*8)-36=
(9*9)-56=
(7*8)*6=
(5*7)*3=
(4*8)*7=
(8*9):2=

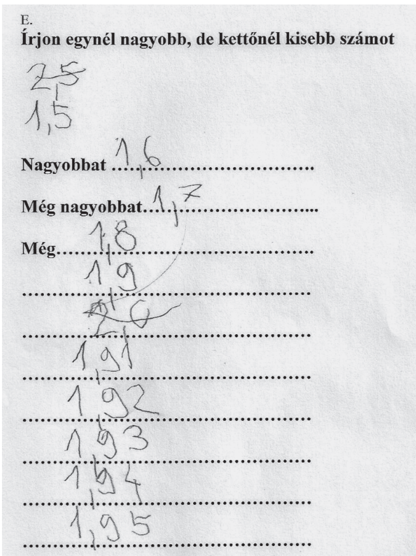
VI. Double bracketing tasks


3*((9+21)+2) =
7*((8+4)+6) =
2*((3*4)+10) =
6*((8*5)+4) =
50-((4+7)*2) =
90-((6+2) *3) =
37-((4+8)*3) =
45-((3+8)*2) =

VII. Generation of bracketing tasks

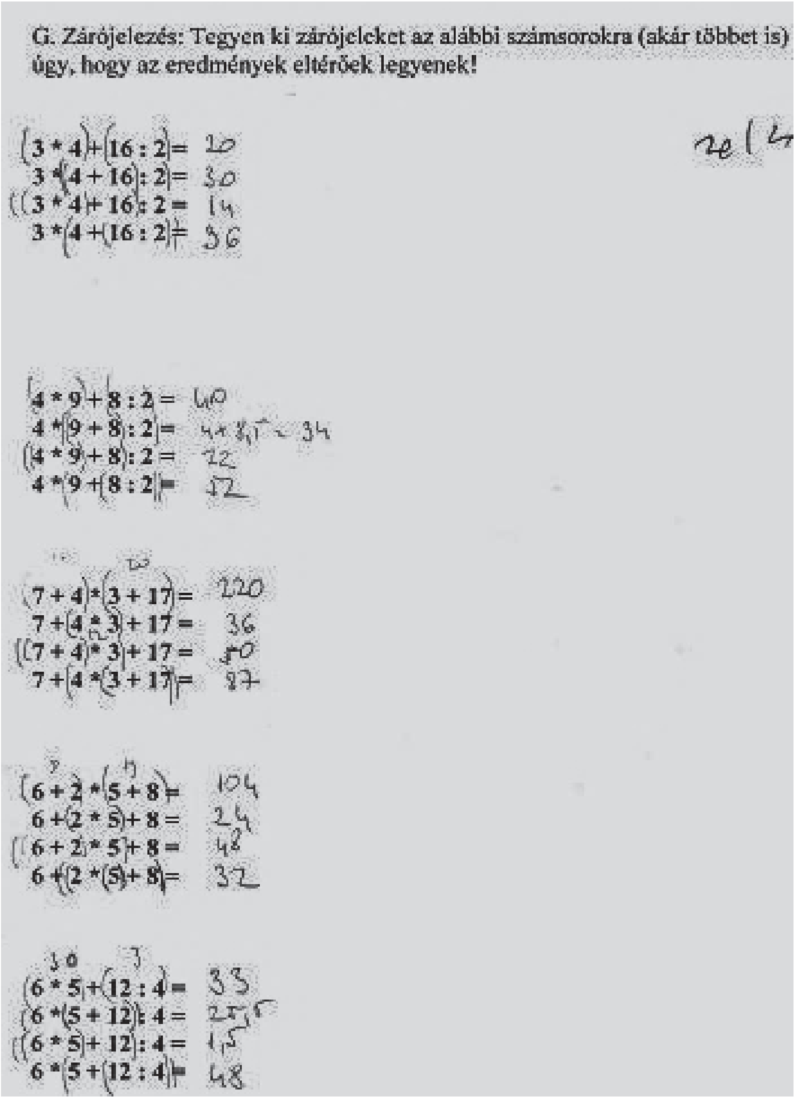


VIII. “Infinity” task



Normal participant: some examples

IX. Generation and resolution of bracketing task


G. Zárójelezés: Tegyen ki zárójeleket az alábbi számsorokra (akár többet is) úgy, hogy az eredmények eltérőek legyenek!

(3 * 4)+(16 : 2)= 20
3 *(4 + 16): 2)= 30
((3 * 4)+ 16): 2 = 14
3 *(4 +(16 : 2))= 36

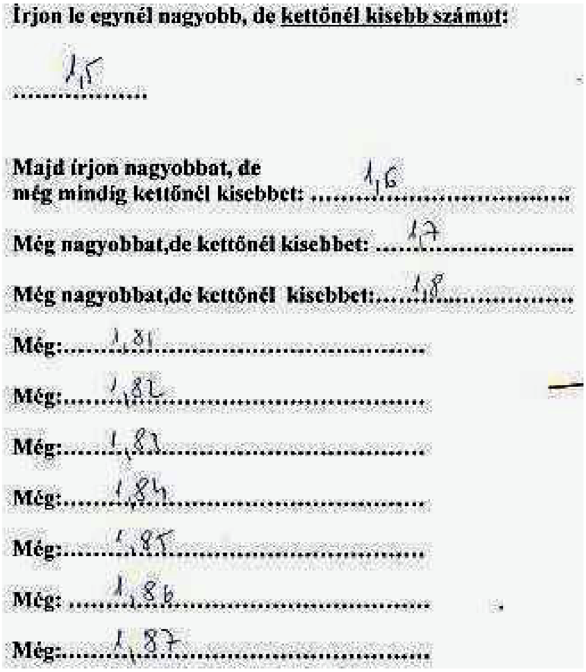
(4 * 9)+(8 : 2)= 40
4 *(9 + 8): 2)= 4×8,5 = 34
((4 * 9)+ 8): 2 = 22
4 *(9 +(8 : 2))= 52

(7 + 4)*(3 + 17)= 220
7 +(4 * 3)+ 17 = 36
((7 + 4)* 3)+ 17 = 50
7 +(4 *(3 + 17))= 87

(6 + 2 *(5 + 8))= 104
6 +(2 * 5)+ 8 = 24
((6 + 2)* 5)+ 8 = 48
6 +(2 *(5)+ 8)= 32

(6 * 5)+(12 : 4)= 33
(6 *(5 + 12)): 4 = 25,5
((6 * 5)+ 12): 4 = 1,5
6 *(5 +(12 : 4))= 48

X. “Infinity” task



Írjon le egynél nagyobb, de kettőnél kisebb számot:

1,5

Majd írjon nagyobbat, de
még mindig kettőnél kisebbet: 1,6

Még nagyobbat,de kettőnél kisebbet: 1,7

Még nagyobbat,de kettőnél kisebbet: 1,8

Még: 1,81

Még: 1,82

Még: 1,83

Még: 1,84

Még: 1,85

Még: 1,86

Még: 1,87

Acknowledgements

The authors wish to thank Zita Örley and Mihály Zsitvai for their help in conducting the tests; our thanks also go to Attila Krajcsi and to the subjects participating in the experiments reported here.

References

Ansari, D., C. Donlan, M. S. C. Thomas, S. A. Ewing, T. Peen and A. Karmiloff-Smith. 2003. What makes counting count? Verbal and visual-spatial contributions to typical and atypical number development. Journal of Experimental Child Psychology 85. 50–62.

Bánréti, Z., I. Hoffmann and V. Vincze. 2016. Recursive subsystems in aphasia and Alzheimer’s disease: Case studies in syntax and theory of mind. Frontiers in Psychology 7. 405.

Butterworth, B. 1999. The mathematical brain. London: Macmillan.

Cohen, L., S. Dehaene, F. Chochon, S. Lehericy and L. Naccache. 2000. Language and calculation within the parietal lobe: A combined cognitive, anatomical and fMRI study. Neuropsychologia 38. 1426–1440.

De Renzi, E. and L. A. Vignolo. 1962. The token test: A sensitive test to detect receptive disturbances in aphasics. Brain 85. 665–678.

Dehaene, S., M. Piazza, P. Pinel and L. Cohen. 2003. Three parietal circuits for number processing. Cognitive Neuropsychology 20. 487–506.

Dehaene, S., E. S. Spelke, P. Pinel, R. Stanescu and S. Tsivkin. 1999. Sources of mathematical thinking: Behavioral and brain-imaging evidence. Science 284. 970–974.

Delazer, M., L. Girelli, A. Graná and F. Domash. 2003. Number processing and calculation – Normative data from healthy adults. Clinical Neuropsychology 17. 331–350.

Friederici, A. D., J. Bahlmann, R. Friedrich and M. Makuuchi. 2011. The neural basis of recursion and complex syntactic hierarchy. Biolinguistics 5. 87–104.

Friedrich, R. M. and A. Friederici. 2013. Mathematical logic in the human brain: Semantics. PLoS ONE 8.

Harskamp, N., van and L. Cipolotti. 2001. Selective impairments for addition, subtraction and multiplication. Implications for the organisation of arithmetical facts. Cortex 37. 363–388.

Hauser, M. D., N. Chomsky and W. T. Fitch. 2002. The faculty of language: What is it, who has it, and how did it evolve? Science 298. 1569–1579.

John, A. A., M. Javali, R. Mahale, A. Mehta, P. T. Acharya and R. J. Srinivasa. 2017. Clinical impression and Western Aphasia Battery classification of aphasia in acute ischemic stroke: Is there a discrepancy? Journal of Neurosciences in Rural Practice 8. 74–78.

Kertesz, A. 1982. The Western Aphasia Battery. New York: Grune & Stratton.

Krajcsi, A. 2006. Enumerating objects: The cause of subitizing and the nature of counting. Budapest: Eötvös Loránd University.

Lemer, C., S. Dehaene, E. Spelke and L. Cohen. 2003. Approximate quantities and exact number words: Dissociable systems. Neuropsychologia 41. 1942–1958.

Osmanné Sági, J. 1991. Az afázia diagnózisa és klasszifikációja [The diagnosis and classification of aphasia]. Ideggyógyászati Szemle 44. 339–362.

Osmanné Sági, J. 1994. A De Renzi–Vignolo beszédmegértési teszt adaptációjának eredményei [The results of adaptation of the comprehension test by De Renzi and D. Vignolo]. Ideggyógyászati Szemle 52. 300–332.

Pesenti, M., M. Thioux, X. Seron and A. De Volder. 2000. Neuroanatomical substrate of Arabic number processing, numerical comparison and simple addition: A PET study. Journal of Cognitive Neuroscience 12. 461–479.

Spelke, E. S. and S. Tsivkin. 2001. Language and number: A bilingual study. Cognition 78. 45–88.

Varley, R. A., N. J. C. Klessinger, C. A. J. Romanowski and M. Siegal. 2005. Agrammatic but numerate. Proceedings of the National Academy of Sciences of the United States of America 102. 3519–3524.

Zago, L., M. Pesenti, E. Mellet, F. Crivello, B. Mazoyer and N. Tzourio-Mazoyer. 2001. Neural correlates of simple and complex mental calculation. Neuroimage 13. 314–327.

Zimmerer, V. and R. A. Varley. 2010. Recursion in severe agrammatism. In H. van der Hulst (ed.) Recursion and human language (Studies in Generative Grammar 104). Berlin: De Gruyter Mouton. 393–405.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 79–104

Morphisms of context-free grammars

TIBOR BEKE
University of Massachusetts, Lowell
tibor_beke@uml.edu

KEYWORDS

context-free grammar
transducer
morphism
category

ABSTRACT

We give a definition of *morphism* from one context-free grammar to another. Thinking of a context-free grammar as the presentation of an algebraic structure (a context-free language) by generators and relations, a morphism of context-free grammars is analogous to a homomorphism from one group (specified by generators and relations) to another. A morphism of grammars induces a mapping between parse trees; for unambiguous context-free grammars, it induces a mapping between well-formed words. This captures the notion of *translation scheme* familiar from the theory of compilers. The composite of two morphisms is a morphism, and context-free grammars and their morphisms form a category. The proof of this – the verification of associativity – is combinatorially quite intricate. We close with some thoughts on syntax-driven translations that are *not* defined by morphisms in the sense given in this paper.

1. Introduction

Let me begin with a little known comment by Noam Chomsky (see Chomsky 1982, 15 or Chomsky 2004, 42), made in response to a question on the significance of automata theory for linguistics and mathematics:

> "This seems to me what one would expect from applied mathematics, to see if you can find systems that capture some of the properties of the complex system that you are working with, and to ask whether those systems have any intrinsic mathematical interest, and whether they are worth studying in abstraction. And that has happened exactly at one level, the level of context-free grammar. At any other level it has not happened. The systems that capture other properties of language, for example, of transformational grammar, hold no interest for mathematics. But I do not think that is a necessary truth. It could turn out that there would be richer and more appropriate mathematical ideas that would capture other, maybe deeper properties of language than context free grammars do. In that case you have another branch of applied mathematics which might have linguistic consequences. That would be exciting."

Let me hasten to say that I do *not* wish to argue with Chomsky's assessment. It would be hard to do, at any rate, since he leaves room for both possibilities: that there is no linguistic theory beyond context-free grammars that is of interest to mathematics; or perhaps there *is*. But I was particularly struck by the sentence *The systems that capture other properties of language, for example, of transformational grammar, hold no interest for mathematics.* From the 1970's on, transformational grammar has been responsible, directly on indirectly, for much research on generalizations of automata that, instead of transforming strings to strings, transform trees to trees. Rational transducers, for example, gave rise to a variety of tree transducers (deterministic, nondeterministic, top-down, bottom-up), in no small part motivated by the desire to find a compact mathematical formalism underlying transformational grammar. In fact, transformations of parse trees, called translations in the computer science literature, are central to the contemporary theory of compilers. There has been a subtle change of perspective, though. Transformational grammar, motivated by examples such as the English passive, seeks to understand operations on tree-like structures within *one* given language. Compilers translate from source code to object code: from one (typically context-free) language to another.

One can appeal to an algebraic analogy at this point. If context-free languages are like algebras, then context-free grammars are like presentations of algebras via generators and relations. One can map one set of generators into another in a way that preserves relations; such a mapping induces a homomorphism of algebras. So there ought to be such a thing as mapping one context-free grammar into another in a 'structure-preserving' way, and this should induce a homomorphism between languages.

The goal of this note is to give *one* possible definition of morphism of context-free grammars. This notion will organize context-free grammars into a category (Mac Lane 1978) in such a way that the effects of morphisms on parse trees – these are, more or less, the 'translations' of computer science – become functorial. The appearance of these category-theoretic concepts is somewhat auxiliary, however, to the main enterprise, which is to understand what it means to map one grammar into another 'in a grammatical way'.

We will be guided by four examples of grammatical operations. Keeping in mind Chomsky's dictum, each of them arises naturally within some body of formalized mathematics – algebra or logic. Going through the motivating examples, the reader is invited to play with the following questions: Which levels of the Chomsky hierarchy do the source and target languages belong to? Which family of transformations (translations? transductions?) does the operation belong to? Each of the motivating examples is given by an explicit formal recipe. Isn't that recipe an outright 'morphism'?

Motivating examples.

(a) In a (non-commutative) ring, the commutator $[x, y]$ of two elements x, y is defined by

$$[x, y] = x \cdot y - y \cdot x$$

Let L_0 be the language of well-formed iterated commutators of elements, and let L_1 be the language of well-parenthesized terms in the function symbols $\cdot$ and $-$. Consider the operation that associates to an expression in L_0 its equivalent in L_1 (prior to expansion and simplification). For example, $[[x, y], z]$ is to be mapped to

$$(((x \cdot y) - (y \cdot x)) \cdot z) - (z \cdot ((y \cdot x) - (x \cdot y))) \ .$$

(b) Consider the language L_0 of (ambiguous) parenthesis-free terms formed from a set of variables with the binary operators $\circledast$ and $\boxtimes$. Let L_1 be the language of terms, with the same operators, in prefix form. Consider the multi-valued mapping that associates to a term in L_0 its prefix forms, under all possible parses. For example, the possible parses of $x \circledast y \boxtimes z$ are (using parentheses, informally)

$$(x \circledast y) \boxtimes z \qquad \text{resp.} \qquad x \circledast (y \boxtimes z)$$

or $\boxtimes \circledast xyz$ resp. $\circledast x \boxtimes yz$ in prefix form.

Can this multi-valued mapping be described without mentioning prefix and infix traversals of binary trees?

(c) Fix a first order signature, and consider the language L of well-formed formulas of first order logic. Let x be a variable, $\mathtt{t}$ a term and ϕ a formula in L. Define the result $\tau_{x \to \mathtt{t}}(\phi)$ of *replacing the free occurrences of* x *in* ϕ *by* $\mathtt{t}$ by the usual set of rules. (These rules will not be recalled here; see e.g., Mendelson 2010 or any careful textbook of logic.) Fix x and $\mathtt{t}$, and consider the map from L to itself sending ϕ to $\tau_{x \to \mathtt{t}}(\phi)$.

(d) Consider again the language L of first order logic. The *negation normal form*, NNF(ϕ) of a formula ϕ is defined by the rewrite rules

$$\neg\neg\phi \Rightarrow \phi$$

$$\neg(\phi \wedge \psi) \Rightarrow \neg\phi \vee \neg\psi$$

$$\neg(\phi \vee \psi) \Rightarrow \neg\phi \wedge \neg\psi$$

$$\neg\forall x\phi \Rightarrow \exists x\neg\psi$$

$$\neg\exists x\phi \Rightarrow \forall x\neg\psi$$

Iterated application of these rules transforms any well-formed formula into a logically equivalent one where the targets of negation symbols (if any) are atomic formulas. Does the operation sending ϕ to NNF(ϕ) belong in the same family as any of (a), (b) or (c)?

Notation. We will consider alphabets $\mathcal{A}$ and context-free grammars G with productions written $x \to \mathsf{s}$ where $x \in \mathcal{A}$ and s is a string in $\mathcal{A}^*$. Neither $\mathcal{A}$ nor G is assumed finite. An element x of $\mathcal{A}$ is non-terminal if it occurs on the left-hand side of some production, and is terminal otherwise. N and T will denote the set of non-terminal and terminal symbols, respectively; so $\mathcal{A} = N \sqcup T$. For $\mathsf{u}, \mathsf{v} \in \mathcal{A}^*$, write $\mathsf{u} \Rightarrow \mathsf{v}$ if v is immediately derivable from u; let $\Rightarrow^+$ denote the transitive and $\Rightarrow^*$ the reflexive-transitive closure of the relation $\Rightarrow$.

We will find it convenient to consider each non-terminal as a possible start symbol, and to consider strings both in the full alphabet $\mathcal{A}$ and in the set of terminals T. For $x \in N$, define

$$\hat{L}_G(x) = \{\mathsf{u} \in \mathcal{A}^* \mid x \Rightarrow^* \mathsf{u}\}$$

and

$$L_G(x) = \{\mathsf{u} \in T^* \mid x \Rightarrow^* \mathsf{u}\}$$

Thus, for non-terminal x, $\hat{L}_G(x)$ is the set of sentential forms that can be generated from x (considered as a start symbol), and $L_G(x)$ is the usual language generated from x.

Let us recall the notion of unambiguous grammar in the form that will be most useful to us:

Definition 1.1. The context-free grammar G is unambiguous if for every non-terminal x and $\mathsf{u} \in \mathcal{A}^*$ with $x \Rightarrow^+ \mathsf{u}$ there exists exactly one pair of k-tuples

$$s_1, s_2, s_3, \ldots, s_k;\ \mathsf{u}_1, \mathsf{u}_2, \mathsf{u}_3, \ldots, \mathsf{u}_k$$

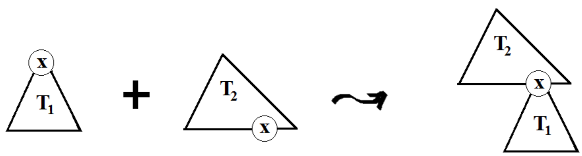
where $s_i \in \mathcal{A}$ and $\mathsf{u}_i \in \mathcal{A}^*$, such that

- $x \to s_1 s_2 s_3 \dots s_k$ is a production
- $\mathsf{u} = \mathsf{u}_1 \mathsf{u}_2 \dots \mathsf{u}_k$, and
- $s_i \Rightarrow^* \mathsf{u}_i$ for each $1 \leqslant i \leqslant k$.

This is equivalent to the requirement that the parse tree of every sentential form $\mathsf{u} \in \hat{L}_G(x)$ be unique; or, equivalently, that there exist a unique leftmost derivation, starting from x, for each $\mathsf{u} \in \hat{L}_G(x)$. If every non-terminal is productive, that is, $L_G(x)$ is non-empty for all non-terminals x, then Def. 1.1 is equivalent to the unambiguity of the $L_G(x)$ in the classical sense. However, Def. 1.1 makes sense even if some or all of the $L_G(x)$ are empty.

Definition 1.2. For $x \in N$, let $\text{tree}_G(x)$ denote the set of parse trees of sentential forms from $\hat{L}_G(x)$, with root x. (One could just as well consider the set of leftmost or rightmost derivations, or other representatives of equivalence classes of derivations, but the formalism of trees is the handiest.) The depth of a tree is the number of nodes on the longest path from root to any leaf, minus 1. Thus, for $T \in \text{tree}_G(x)$, $\text{depth}(T) = 0$ if and only if T consists solely of the root (which is also a leaf) x. Note that $\text{depth}(T) = 1$ if and only if T equals some production $x \to \mathsf{s} \in G$. Let $\textsc{nt}(T)$ denote the set of leaves of T labeled by non-terminal symbols; for a node t of T, let $\text{label}(t)$ denote the label (i.e., element of the alphabet $\mathcal{A}$) at t.

Let $T_1 \in \text{tree}_G(x)$ and let T_2 be a tree with a leaf t such that $\text{label}(t) = x$. We will skip the definition of the horticultural maneuver of *grafting* T_1 *onto* T_2 *at the location* t. It is the same as the composition of (chains of) productions, as the illustration(s) below will make it clear.



2. Morphisms of grammars

Let G_0 and G_1 be context-free grammars in the alphabets $\mathcal{A}_0$ and $\mathcal{A}_1$, with terminals T_0, T_1 and non-terminals N_0, N_1 respectively.

Definition 2.1. A *morphism from* G_0 *to* G_1 consists of the following data:

- a mapping $\alpha : N_0 \to N_1$
- a mapping β that assigns to each production $x \to \mathsf{s} \in G_0$ an element of $\text{tree}_{G_1}(\alpha(x))$
- for each production $p \in G_0$, a function $\gamma(p, -)$ from $\textsc{nt}(\beta(p))$ to $\textsc{nt}(p)$, with the property that for all $t \in \textsc{nt}(\beta(p))$,

$$\alpha\big(\text{label}(\gamma(p,t))\big) = \text{label}(t)\ .$$

More plainly, α gives the translation of lexical categories. β specifies, for each production $p : x \to \mathsf{s}$ in the source grammar, a parse tree in the target grammar, with root $\alpha(x)$. Productions of the form $x \to \mathsf{s}$ will be translated to trees of the form $\beta(x \to \mathsf{s})$. The re-indexing map $\gamma(p, -)$ associates to the location of each non-terminal symbol r occurring as a leaf in $\beta(x \to \mathsf{s})$ the location of a non-terminal symbol s in s such that α will translate s to r. This permits translation of the input parse tree by either top-down or bottom-up recursion.

Let us make this more concrete by a formalization of our motivating example (a). For the sake of readability, we will depart from the BNF convention of enclosing names of non-terminals in angle brackets; strings typeset in sans serif font, such as var and expr, should be considered as stand-alone symbols. Also, we will drop commas separating elements of a set being listed. Dots '...' indicate a (potentially infinite) set indexed by the natural numbers.

Example 2.2. Consider the source alphabet

$$N_0 = \{\ \mathsf{var}\ \mathsf{expr}\ \}$$
$$T_0 = \{\ [\ ,\ ]\ x_1\ x_2\ \dots\ x_i\ \dots \}$$

Let the grammar G_0 consist of the productions

$$\begin{aligned} \mathsf{var} &\to x_1 \mid x_2 \mid \ldots \mid x_i \mid \ldots \\ \mathsf{expr} &\to [\mathsf{var}, \mathsf{var}] \\ \mathsf{expr} &\to [\mathsf{var}, \mathsf{expr}] \\ \mathsf{expr} &\to [\mathsf{expr}, \mathsf{var}] \\ \mathsf{expr} &\to [\mathsf{expr}, \mathsf{expr}] \end{aligned}$$

Now consider the target alphabet

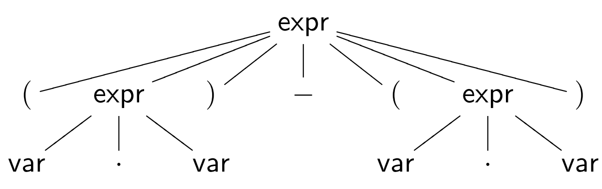
$$\begin{aligned} N_1 &= \{ \ \mathsf{var} \ \mathsf{expr} \ \} \\ T_1 &= \{ \ (\,) \ - \ \cdot \, x_1 \, x_2 \, \ldots \, x_i \, \ldots \} \end{aligned}$$

Let the grammar G_1 consist of the productions

$$\begin{aligned} \mathsf{var} &\to x_1 \mid x_2 \mid \ldots \mid x_i \mid \ldots \\ \mathsf{expr} &\to \mathsf{var} - \mathsf{var} \mid \mathsf{var} \cdot \mathsf{var} \\ \mathsf{expr} &\to \mathsf{var} - (\mathsf{expr}) \mid \mathsf{var} \cdot (\mathsf{expr}) \\ \mathsf{expr} &\to (\mathsf{expr}) - \mathsf{var} \mid (\mathsf{expr}) \cdot \mathsf{var} \\ \mathsf{expr} &\to (\mathsf{expr}) - (\mathsf{expr}) \mid (\mathsf{expr}) \cdot (\mathsf{expr}) \end{aligned}$$

There is a morphism from G_0 to G_1 with components α, β, γ defined by

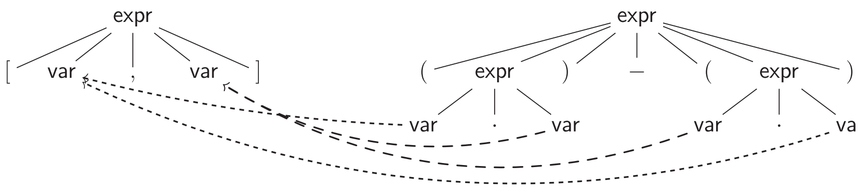
- $\alpha(\,\mathsf{expr}\,) = \mathsf{expr}$ and $\alpha(\,\mathsf{var}\,) = \mathsf{var}$
- $\beta(\,\mathsf{var} \to x\,) = x$ for any variable x; note that $\gamma(\mathsf{var} \to x, -)$ has empty domain
- $\beta(\,\mathsf{expr} \to [\mathsf{var}, \mathsf{var}]\,)$ is



generating the string $(\mathsf{var} \cdot \mathsf{var}) - (\mathsf{var} \cdot \mathsf{var})$. Let us refer to the leaves of the above tree via their location in '$(\mathsf{var} \cdot \mathsf{var}) - (\mathsf{var} \cdot \mathsf{var})$'; so the leaves labeled with non-terminals occur at $\{2, 4, 8, 10\}$. Similarly, let us refer to the leaves in $\textsc{nt}(\,\mathsf{expr} \to [\mathsf{var},\mathsf{var}]\,)$ through their location in the string '$[\mathsf{var},\mathsf{var}]$', i.e., $\{2,4\}$. Then define

$$\begin{aligned}\gamma(\mathsf{expr} \to [\mathsf{var},\mathsf{var}],2) &= 2\\ \gamma(\mathsf{expr} \to [\mathsf{var},\mathsf{var}],4) &= 4\\ \gamma(\mathsf{expr} \to [\mathsf{var},\mathsf{var}],8) &= 4\\ \gamma(\mathsf{expr} \to [\mathsf{var},\mathsf{var}],10) &= 2\end{aligned}$$

Visually, the re-indexing map $\gamma(\mathsf{expr} \to [\mathsf{var},\mathsf{var}],-)$ is indicated by the dotted and broken arrows



Continuing with the next production, define

$$\beta(\,\mathsf{expr} \to [\mathsf{var},\mathsf{expr}]\,) = (\mathsf{var}\cdot(\mathsf{expr})) - ((\mathsf{expr})\cdot\mathsf{var})$$

(Since G_1 is unambiguous, we will identify sentential forms with their parse trees.) Using the same coding of locations as above, define

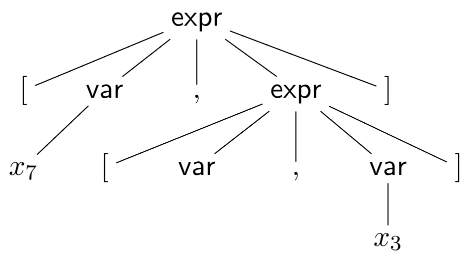
$$\begin{aligned}\gamma(\mathsf{expr} \to [\mathsf{var},\mathsf{expr}],2) &= 2\\ \gamma(\mathsf{expr} \to [\mathsf{var},\mathsf{expr}],5) &= 4\\ \gamma(\mathsf{expr} \to [\mathsf{var},\mathsf{expr}],10) &= 4\\ \gamma(\mathsf{expr} \to [\mathsf{var},\mathsf{expr}],13) &= 2\end{aligned}$$

The treatment of the other two productions, and re-indexing of non-terminals therein, is analogous.

How does translation from $\hat{L}_{G_0}(\mathsf{expr})$ to $\hat{L}_{G_1}(\mathsf{expr})$ actually work? Consider a sentential form generated by G_0 from expr, say,

$$[x_7,[\mathsf{var},x_3]]$$

with parse tree



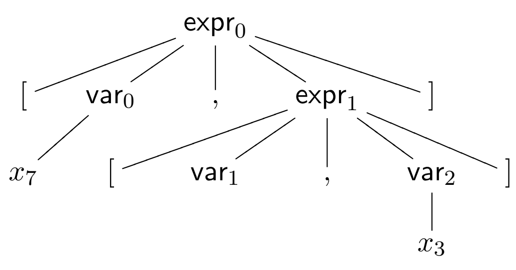
Since G_1 is unambiguous, the process is easiest to describe by bottom-up induction. Starting from the leaves, associate to each non-terminal symbol t in the input tree a string $\tau(t)$ from $\hat{L}_{G_1}(\alpha(x))$:

- If t is a leaf, let $\tau(t) = \alpha(t)$.
- If t is var, with descendant $\mathsf{var} \to x$, set $\tau(\mathsf{var}) = x$.
- Suppose t is a node expr with descendants, say, $[\mathsf{var}, \mathsf{var}]$. Let $s_1 = \tau(\mathsf{var})$ for the first occurrence of 'var' in $[\mathsf{var}, \mathsf{var}]$, and $s_2 = \tau(\mathsf{var})$ for the second occurrence. (τ is supposed to be defined on those two symbols by induction.) Then set

  $$\tau(\mathsf{expr}) = (s_1 \cdot s_2) - (s_2 \cdot s_1)$$

  The idea is analogous for the other productions with source expr.

To see what is going on, let us affix subscripts to the non-terminals of the above parse tree:



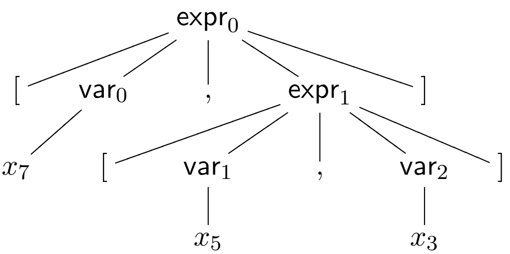
Then

$$\tau(\mathsf{var}_1) = \mathsf{var} \qquad \tau(\mathsf{var}_0) = x_7 \qquad \tau(\mathsf{var}_2) = x_3$$

$$\tau(\mathsf{expr}_1) = (\mathsf{var} \cdot x_3) - (x_3 \cdot \mathsf{var})$$

$$\tau(\mathsf{expr}_0) = (x_7 \cdot ((\mathsf{var} \cdot x_3) - (x_3 \cdot \mathsf{var}))) - (((\mathsf{var} \cdot x_3) - (x_3 \cdot \mathsf{var})) \cdot x_7)$$

For the parse tree



a moment's thought confirms that

$$\tau(\mathsf{expr}_0) = (x_7 \cdot ((x_5 \cdot x_3) - (x_3 \cdot x_5))) - (((x_5 \cdot x_3) - (x_3 \cdot x_5)) \cdot x_7)$$

respecting all long-distance dependencies.

Above, G_0 was an unambiguous grammar, hence one could talk of the translation of a string or of a parse tree interchangeably. The next proposition defines the effect of a morphism of grammars in general. We retain the notation of Def. 2.1.

Proposition 2.3. *A morphism of grammars from G_0 to G_1 induces, for each $x \in N_0$, a mapping*

$$\tau : \mathrm{tree}_{G_0}(x) \to \mathrm{tree}_{G_1}(\alpha(x)) .$$

Indeed, for $T \in \mathrm{tree}_{G_0}(x)$, define $\tau(T) \in \mathrm{tree}_{G_1}(\alpha(x))$ by induction on the depth of T:

- If $\mathrm{depth}(T) = 0$, then T must be x itself, and $\tau(T)$ is defined to be $\alpha(x)$.
- If $\mathrm{depth}(T) > 0$, let $x \to \mathsf{s} \in G_0$ be the top production in T. Write p for $x \to \mathsf{s}$ for brevity. Note that $\mathrm{NT}(p)$ can be identified with a subset of s, namely, the locations of the non-terminal symbols in s. Since G_0 is context-free, each $s \in \mathrm{NT}(p)$ induces a subtree T_s of T with s as root. For each $t \in \mathrm{NT}(\beta(p))$, graft the tree $\tau(T_{\gamma(p,t)})$ on $\beta(p)$ with t as root. $\tau(T)$ is defined to be the resulting tree.

The definition makes sense: since $\mathrm{depth}(T_s) < \mathrm{depth}(T)$ for any $s \in \mathrm{NT}(p)$, $\tau(T_s)$ is defined by the induction hypothesis. Note that $\tau(T_s)$ belongs to $\mathrm{tree}_{G_1}(\alpha(\mathrm{label}(s))$ by the induction assumption, and $\alpha\big(\mathrm{label}(\gamma(p,t))\big) = \mathrm{label}(t)$ by Def. 2.1. That is, the non-terminal symbol at the root of

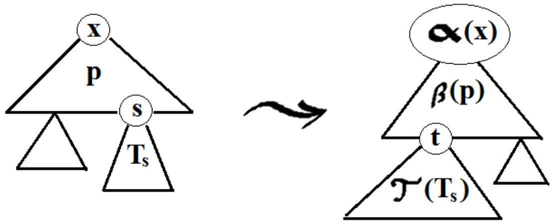


Figure 1: Computing $\tau(T)$. Above, p is a production (i.e., tree of depth 1), $\beta(p)$ is a tree, s and t are leaves labeled with non-terminal symbols such that $s = \gamma(p,t)$. x and $\alpha(x)$ are the labels of the roots.

$\tau(T_{\gamma(p,t)})$ coincides with the non-terminal symbol at the location t. Since G_1 is a context-free grammar, the graft is well-defined, and $\tau(T)$ will belong to $\text{tree}_{G_1}(\alpha(x))$ as desired. □

Obviously, one can rewrite the above recursive definition into an algorithm to compute $\tau(T)$ by bottom-up induction on T, from leaves toward the root. Note that if $\text{depth}(T) = 1$, that is, T is a production $x \to \mathsf{s}$ in G_0, then $\tau(T)$ ends up being the same as $\beta(T)$.

We will sometimes consider the induced translation τ as a multi-valued mapping

$$\hat{L}_{G_0}(x) \to \hat{L}_{G_1}\big(\alpha(x)\big) \ .$$

Indeed, for each $\mathsf{u} \in \hat{L}_{G_0}(x)$, there is a value for each parse tree T of u, namely, the string in $\hat{L}_{G_1}\big(\alpha(x)\big)$ generated by $\tau(T)$.

Proposition 2.4. *For any $x \in N_0$ and $\mathsf{u} \in L_{G_0}(x)$ with parse tree T, $\tau(T)$ generates a string in $L_{G_1}\big(\alpha(x)\big)$.*

Proof. By induction on the depth of T. $\text{depth}(T) = 0$ is impossible, since x is assumed non-terminal and u is a string of terminals. If $\text{depth}(T) = 1$, then T consists of the single production $x \to \mathsf{u} \in G_0$. The leaves of $\tau(x \to \mathsf{u}) = \beta(x \to \mathsf{u})$ must consist of terminals. Indeed, if there was a leaf labeled with a non-terminal, then $\gamma(x \to \mathsf{u}, -)$ would need to map its location to the location of some non-terminal in u, but u does not contain any non-terminals. So $\tau(T) = \beta(x \to \mathsf{u})$ generates a string in $L_{G_1}\big(\alpha(x)\big)$.

If $\text{depth}(T) > 1$, then $\tau(T)$ is, by the definition, the result of grafting trees of the form $\tau(T_s)$, for subtrees T_s of T, onto those leaves of $\beta(x \to \mathsf{s})$ that contain non-terminals. Using the induction hypothesis, all leaves of $\tau(T_s)$ are labeled with terminal symbols; hence $\tau(T)$ generates an element of $L_{G_1}\big(\alpha(x)\big)$ as well. □

Let us summarize the discussion so far:

Corollary 2.5. *A morphism* (α, β, γ) *of context-free grammars from* G_0 *to* G_1 *induces a function, for each* $x \in N_0$*, from* $\mathrm{tree}_{G_0}(x)$ *to* $\mathrm{tree}_{G_1}(\alpha(x))$*. This induces, in turn, a multi-valued function from* $\hat{L}_{G_0}(x)$ *to* $\hat{L}_{G_1}\big(\alpha(x)\big)$*, which restricts to a multi-valued function from* $L_{G_0}(x)$ *to* $L_{G_1}\big(\alpha(x)\big)$*. If* G_0 *is an unambiguous grammar, then the latter two maps are single-valued.*

Example 2.6. Returning to our motivating example (b), consider the source alphabet

$$N_0 = \{\ \mathsf{expr}\ \}$$
$$T_0 = \{\ \circledast\ \boxtimes\ x_1\, x_2\ \ldots\ x_i\ \ldots\}$$

Let the grammar G_0 consist of the productions

$$\mathsf{expr} \to x_1 \mid x_2 \mid \ \ldots\ \mid x_i \mid \ \ldots$$
$$\mathsf{expr} \to \mathsf{expr} \circledast \mathsf{expr}$$
$$\mathsf{expr} \to \mathsf{expr} \boxtimes \mathsf{expr}$$

Now consider the target grammar G_1 with identical alphabet $N_1 = N_0$, $T_1 = T_0$ but productions

$$\mathsf{expr} \to x_1 \mid x_2 \mid \ \ldots\ \mid x_i \mid \ \ldots$$
$$\mathsf{expr} \to \circledast\, \mathsf{expr}\, \mathsf{expr}$$
$$\mathsf{expr} \to \boxtimes\, \mathsf{expr}\, \mathsf{expr}$$

There is a morphism from G_0 to G_1 with components α, β, γ defined by

- $\alpha(\,\mathsf{expr}\,) = \mathsf{expr}$
- $\beta(\,\mathsf{expr} \to x\,) = x$ for any variable x; note that $\gamma(\mathsf{expr} \to x, -)$ has empty domain
- $\beta(\,\mathsf{expr} \to \mathsf{expr} \circledast \mathsf{expr}\,) = \circledast\, \mathsf{expr}\, \mathsf{expr}$ with $\gamma(2) = 1$ and $\gamma(3) = 3$
- $\beta(\,\mathsf{expr} \to \mathsf{expr} \boxtimes \mathsf{expr}\,) = \boxtimes\, \mathsf{expr}\, \mathsf{expr}$ with $\gamma(2) = 1$ and $\gamma(3) = 3$

(Since G_1 is unambiguous, there is no loss in writing the values of β as strings, as opposed to parse trees. The first argument of γ is suppressed for the sake of readability; numbers refer to locations of non-terminal symbols, as before.) For any $\mathsf{u} \in L_{G_0}(\mathsf{expr})$, the values of $\tau(\mathsf{u})$ will be the prefix forms of the parses of u.

Before moving on to compositions of morphisms and the rest of our motivating examples, let us make a series of remarks.

• The definition of morphism of grammars, as given above, appears out of the blue, and in somewhat austere generality. Admittedly, the definition, like most in the realm of algebra, is 'experimental', and driven by several, not easily formalizable criteria. It should cover enough cases of interest, seemingly not otherwise connected; it should possess good structural properties; and should have a family, or conceptual resemblance to other notions that have proved useful. As for the instances of morphisms of grammars in mathematical syntax, I am hopeful this article provides quite a few. The desired structure theory is phrased in the language of categories; see below. As for family resemblances, there exist significant overlaps between the formalisms of tree transducers, term rewrite systems and context-free language transformations, discussion of which would take us far afield. Suffice it to say that the notion of morphism of grammars is most similar to (and in fact, properly contains) *synchronous context-free grammars* (SCFG); see e.g., Chapter 23 of Atallah & Blanton (2010). SCFG are themselves notational variants of the *syntax-directed translation schemata* of Aho & Ullman (1972). The differences are quite significant:

– unlike SCFG, morphisms assume the existence of source and target grammars, their alphabets linked by a map α

– SCFG pair rules with rules; morphisms associate to each rule in the source grammar a parse tree in the target grammar

– in a SCFG, each re-indexing map is a permutation of non-terminal symbols; in a morphism, the re-indexing datum $\gamma(x \to \mathsf{s}, -)$ is a map from the *locations* of non-terminal symbols in $\beta(x \to \mathsf{s})$ to the *locations* of non-terminal symbols in s.

Thus, because of the presence of repeated variables, our motivating example (a) could not be handled by a SCFG. Nonetheless, it is fair to think of morphisms of grammars as syntax-directed translation schemes, boosted to their 'natural level of generality'.

• Recall that our grammars do not contain preferred start symbols; a morphism of grammars induces a multi-valued map

$$\hat{L}_{G_0}(x) \to \hat{L}_{G_1}\big(\alpha(x)\big)$$

for each non-terminal x in the alphabet $\mathcal{A}_0$ of G_0. It may well happen that for some $\mathsf{u} \in \mathcal{A}_0^*$, there exist distinct $x_0, x_1 \in N_0$ such that $\mathsf{u} \in \hat{L}_{G_0}(x_0)$ and

$\mathsf{u} \in \hat{L}_{G_0}(x_1)$, and the translation(s) into $\hat{L}_{G_1}$ differ when u is considered as a descendant of x_0 from when it is considered a descendant of x_1.

• The language of iterated commutators, cf. Example 2.2, could be more succinctly defined with the help of a single non-terminal symbol expr and productions

$$\mathsf{expr} \to x_1 \mid x_2 \mid \ldots \mid x_i \mid \ldots$$
$$\mathsf{expr} \to [\mathsf{expr}, \mathsf{expr}]$$

However, as long as one prefers to put parentheses around compound expressions, but not around individual variables in the language of terms with infix operators $-$ and $\cdot$, one needs both of the syntactic categories var and expr in the target language. This, in turn, necessitates that the source language should distinguish variables from compound expressions; hence the more labored grammar G_0 of Example 2.2. This observation highlights that our morphisms are defined between context-free *grammars*, and are sensitive to the choice of grammar, even for unambiguous languages.

• What seems to be conspicuously missing from the definition of morphism is how the terminal symbols get translated. Indeed, the function α that is part of the morphism data goes from non-terminal symbols to non-terminal symbols. Of course, the function β is responsible for the translation of terminals, since terminals occurring in the language can be reached from the source non-terminal via productions. In fact, the reader may enjoy working the following out. Let T_0, T_1 be alphabets. Recall that any map $h : T_0 \to T_1^*$ induces a semigroup homomorphism $h : T_0^* \to T_1^*$. (The reuse of the letter 'h' should cause no confusion.) For a language $L \subseteq T_0^*$, h restricts to a map $h : L \to T_1^*$. Maps of this type are called literal homomorphisms.

Exercise. Let G_0 be a context-free grammar in the alphabet $N_0 \sqcup T_0$ and T_1 another set of terminals. Let $h : T_0 \to T_1^*$ be a map, inducing a literal homomorphism $h : L_{G_0}(x) \to T_1^*$ for each $x \in N_0$. Show that there exists a context-free grammar G_1 in the alphabet $N_0 \sqcup T_1$ and a morphism of grammars $G_0 \to G_1$ whose associated translation $\tau : L_{G_0}(x) \to L_{G_1}(x)$ is single-valued and satisfies $\tau(\mathsf{u}) = h(\mathsf{u})$ for all $\mathsf{u} \in L_{G_0}(x)$, any $x \in N_0$. (Hint: extend h to a semigroup homomorphism $(N_0 \sqcup T_0)^* \to (N_0 \sqcup T_1)^*$ by setting $h(x) = x$ for $x \in N_0$. α is the identity. Now let $\beta(x \to \mathsf{s}) = h(\mathsf{s})$.)

That is, any literal homomorphism can be induced by a morphism of grammars. Similarly, any rational transducer (thought of as a multi-valued mapping from its domain to its range, both being rational languages) can be encoded via a morphism of grammars. The details of this encoding are

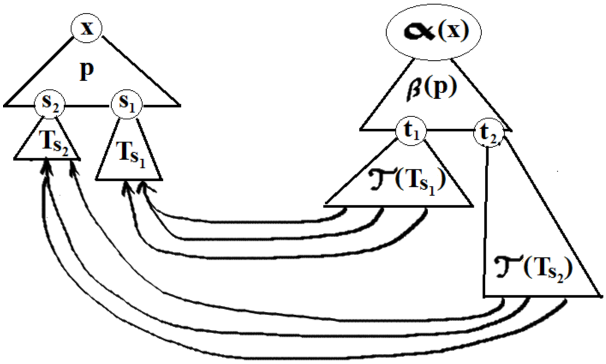


Figure 2: Defining $\gamma(T,-)$. Above, p is a production (i.e., tree of depth 1), $\beta(p)$ is a tree, s_1, s_2, t_1 and t_2 are leaves labeled with non-terminal symbols such that $s_1 = \gamma(p, t_1)$ and $s_2 = \gamma(p, t_2)$.

straightforward, but will be skipped here. It is unlikely that the notion of morphism of grammars will have anything to add to the very fine-tuned theory of rational transducers.

The next proposition is a simultaneous extension of Prop. 2.4 and of the defining property of the re-indexing map γ from the definition of morphism.

Proposition 2.7. *Let (α, β, γ) be a morphism of context-free grammars from G_0 to G_1, $x \in N_0$ and $T \in \text{tree}_{G_0}(x)$. There is a natural map $\gamma(T, -)$ from $\text{NT}(\tau(T))$ to $\text{NT}(T)$ such that for any $t \in \text{NT}(\tau(T))$,*

$$\alpha\big(\text{label}(\gamma(T, t))\big) = \text{label}(t)\ .$$

Proof. By induction on the depth of T. If $\text{depth}(T) = 0$ then T consists of just the root $x \in N_0$, and $\tau(T)$ is the tree containing only the root $\alpha(x) \in N_1$. So $\text{NT}(T) = \{x\}$ and $\text{NT}(\tau(T)) = \{\alpha(x)\}$; $\gamma(T, -)$ is uniquely determined.

If $\text{depth}(T) > 0$, recall how $\tau(T)$ is defined. Let $p \in G_0$ be the top production in T. As before, this induces subtrees T_s of T with roots $s \in \text{NT}(p)$. For each $t \in \text{NT}(\beta(p))$, graft the tree $\tau(T_{\gamma(p,t)})$ on $\beta(p)$ with t as root. $\tau(T)$ is defined to be the resulting tree.

Consider any $t \in \text{NT}(\beta(p))$ and let $s = \gamma(p, t)$. Since $\text{depth}(T_s) < \text{depth}(T)$, by the induction hypothesis there is a map $\gamma(T_s, -)$ from $\text{NT}(\tau(T_s))$ to $\text{NT}(T_s)$, with α as left inverse to the action of $\gamma(T_s, -)$ on labels. When grafting $\tau(T_s)$ to $\beta(p)$, the domain of $\gamma(T_s, -)$ can be shifted with it, to become a subset of $\text{NT}(\tau(T))$.

However, $\text{NT}(\tau(T))$ is the disjoint union of the various $\text{NT}(\tau(T_s))$ grafted to $\beta(p)$, with $s = \gamma(p,t)$, as t ranges over $\text{NT}(\beta(p))$. $\gamma(T,-)$ can thus be defined as the disjoint union of the (appropriately shifted) maps $\gamma(T_s,-)$. □

Note that if T is a production $x \to \mathsf{s} \in G_0$ then $\gamma(T,-)$, as constructed above, coincides with $\gamma(x \to \mathsf{s},-)$ that is part of the morphism data; there is thus no conflict of notation.

Observe also that when T is a parse tree of some string u containing only terminal symbols then the leaves of $\tau(T)$ cannot contain non-terminals either (since no map $\gamma(T,-)$ with the properties above could exist); so we indeed have an extension of Prop. 2.4.

Our choice of terminology insinuates that morphisms can be composed, and, with context-free grammars as objects, form a category. We will treat this next.

Definition 2.8. Let G_0, G_1, G_2 be context-free grammars, and let $(\alpha_{01}, \beta_{01}, \gamma_{01})$ be a morphism from G_0 to G_1, and $(\alpha_{12}, \beta_{12}, \gamma_{12})$ a morphism from G_1 to G_2. Define their composite

$$(\alpha_{02}, \beta_{02}, \gamma_{02}) = (\alpha_{01}, \beta_{01}, \gamma_{01}) \star (\alpha_{12}, \beta_{12}, \gamma_{12})$$

a morphism from G_0 to G_2, as follows:

$$\alpha_{02} \text{ is the composite } N_0 \xrightarrow{\alpha_{01}} N_1 \xrightarrow{\alpha_{12}} N_2.$$

Let $x \to \mathsf{s}$ (abbreviated as p) be a production in G_0. Set $\beta_{02}(p) = \tau_{12}\big(\beta_{01}(p)\big)$, where τ_{12} is the induced translation from $\text{tree}_{G_1}(y)$ to $\text{tree}_{G_2}\big(\alpha_{12}(y)\big)$, for $y \in N_1$. Note that $\beta_{02}(x \to \mathsf{s})$ is an element of $\text{tree}_{G_2}(\alpha_{12}(\alpha_{01}(x)))$, i.e., of $\text{tree}_{G_2}(\alpha_{02}(x))$, as required.

$\gamma_{02}(p,-)$ is to be a map from $\text{NT}(\beta_{02}(p))$ to $\text{NT}(p)$. It is defined as the composite

$$\text{NT}\big(\tau_{12}(\beta_{01}(p))\big) \xrightarrow{\gamma_{12}(\beta_{01}(p),-)} \text{NT}(\beta_{01}(p)) \xrightarrow{\gamma_{01}(p,-)} \text{NT}(p)$$

More plainly, a production $p \in G_0$ is translated by β_{01} into a parse tree T_1 formed with G_1, which τ_{12} translates into a parse tree T_2 formed with G_2. The re-indexing map $\gamma_{12}(\beta_{01}(p),-)$ goes from leaves of T_2 labeled with non-terminal symbols to leaves of T_1 labeled with non-terminal symbols, followed by the re-indexing map $\gamma_{01}(p,-)$ from leaves of T_1 labeled with non-terminal symbols, to leaves (i.e., letters on the right-hand side) of the production p that are non-terminal symbols.

As a continuation of Example 2.6, it is instructive at this point to construct grammars G_0, G_1, G_2 for prefix resp. postfix resp. fully parenthesized infix terms of binary function symbols $\circledast$ and $\boxtimes$, and morphisms $G_i \to G_j$ $(i, j \in \{0, 1, 2\})$ that form a commutative diagram of isomorphisms. Of course, one expects more: a commutative diagram of (iso)morphisms of grammars should induce a commutative diagram of (bijective) mappings between the associated languages. That is indeed so. To prove it, we need a key structural property of τ. By definition, the translation $\tau(T)$ of a parse tree T can be generated by attaching to the translation of the top production in T the translations of the sub-trees of the top production – appropriately re-indexed. The next lemma states that the same recipe applies if one separates any top segment, not necessarily just the top production, of the input tree. The necessary re-indexing is supplied by Prop. 2.7.

Lemma 2.9. *Let $x \in N_0$ and $T \in \mathrm{tree}_{G_0}(x)$. For each $s \in \mathrm{NT}(T)$, suppose given $U_s \in \mathrm{tree}_{G_0}(\mathrm{label}(s))$. Let $T_U \in \mathrm{tree}_{G_0}(x)$ be the result of grafting each U_s to s as root. Now, for each $t \in \mathrm{NT}(\tau(T))$, graft $\tau(U_{\gamma(T,t)})$ to $\tau(T)$ with t as root. Let $\tau(T)_{\tau(U)} \in \mathrm{tree}_{G_1}(\alpha(x))$ be the resulting tree. Then $\tau(T_U) = \tau(T)_{\tau(U)}$.*

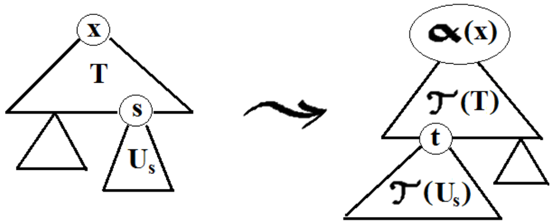


Figure 3: Modularity of $\tau(T)$. s and t are leaves labeled with non-terminal symbols such that $s = \gamma(p, t)$. x and $\alpha(x)$ are labels of the roots. Note the similarity with Fig. 1.

Proof. By induction on depth(T). When depth(T) $= 0$, the lemma is a tautology. When depth(T) $= 1$, it is the inductive step in the definition of τ (applied to the tree T_U, whose top production is T).

If depth(T) > 1, let $p \in G_0$ be the top production in T. As before, this induces subtrees T_r of T with roots $r \in \mathrm{NT}(p)$. The set of leaves of T with non-terminal labels, $\mathrm{NT}(T)$, is the disjoint union of $\mathrm{NT}(T_r)$ as r ranges over $\mathrm{NT}(p)$. For each $r \in \mathrm{NT}(p)$, let $T_{r,U}$ be the tree that results from grafting

U_s to s for each $s \in \text{NT}(T_r)$. $T_{r,U}$ is thus the same as the subtree of T_U with r as root.

$\tau(T_U)$ (by the inductive step in the definition of τ) is the result of grafting $\tau(T_{\gamma(p,v),U})$ to v, for v ranging over $\text{NT}(\beta(p))$. Pick such a $v \in \text{NT}(\beta(p))$ and let $r = \gamma(p, v)$. Since $\text{depth}(T_{r,U}) < \text{depth}(T)$, by the induction hypothesis $\tau(T_{r,U})$ is the same as the result of grafting, for each $t \in \text{NT}(\tau(T_r))$, $\tau(U_{\gamma(T_r,t)})$ to t as root. As v ranges over $\text{NT}(\beta(p))$, this assembles to the same tree as $\tau(T)$ with $\tau(U_{\gamma(T,t)})$ grafted to t for each $t \in \text{NT}(\tau(T))$. But that is the same as $\tau(T)_{\tau(U)}$ by definition, completing the induction step. □

Proposition 2.10. *If G_0, G_1, G_2 are context-free grammars and $(\alpha_{01}, \beta_{01}, \gamma_{01}) : G_0 \to G_1$ resp. $(\alpha_{12}, \beta_{12}, \gamma_{12}) : G_1 \to G_2$ morphisms of grammars, with composite $(\alpha_{02}, \beta_{02}, \gamma_{02}) : G_0 \to G_2$ and associated translation functions τ_{01}, τ_{12} and τ_{02}. Then for all $x \in N_0$ and $T \in \text{tree}_{G_0}(x)$,*

$$\tau_{12}\big(\tau_{01}(T)\big) = \tau_{02}(T) \quad \text{in } \text{tree}_{G_2}(\alpha_{02}(x)) \ .$$

Proof. When $\text{depth}(T) = 0$, this reduces to $\alpha_{12}\big(\alpha_{01}(x)\big) = \alpha_{02}(x)$. When $\text{depth}(T) > 0$, let p be the top production in T, inducing subtrees T_s with roots $s \in \text{NT}(p)$ as before. $\tau_{01}(T)$, by definition, is the result of grafting $\tau_{01}(T_{\gamma_{01}(p,t)})$ to t for each $t \in \text{NT}(\beta_{01}(p))$. τ_{12} of that composite tree, by Lemma 2.9, is the result of grafting $\tau_{12}\big(\tau_{01}(T_{\gamma_{01}(p,t)})\big)$, with $t = \gamma_{12}(\tau_{12}(\beta_{01}(p)), r)$, to $r \in \text{NT}(\tau_{12}(\beta_{01}(p)))$. But that is the same as the translation of T under τ_{02}, by definition of the composite of two morphisms. □

Proposition 2.11. *The composition of morphisms of context-free grammars is associative. That is, if G_i $(i = 0, 1, 2, 3)$ are context-free grammars, and $\mu_{i,i+1} = (\alpha_{i,i+1}, \beta_{i,i+1}, \gamma_{i,i+1})$ morphisms from G_i to G_{i+1} (here $i = 0, 1, 2$) then*

$$\mu_{01} \star (\mu_{12} \star \mu_{23}) = (\mu_{01} \star \mu_{12}) \star \mu_{23} \ .$$

Proof. The component α_{03} of $G_0 \to G_3$ is the composite

$$N_0 \xrightarrow{\alpha_{01}} N_1 \xrightarrow{\alpha_{12}} N_2 \xrightarrow{\alpha_{23}} N_3 \ .$$

As regards β_{03}: given $p \in G_0$, $\mu_{01} \star (\mu_{12} \star \mu_{23})$ associates to it $\tau_{13}\big(\beta_{01}(p)\big)$, while $(\mu_{01} \star \mu_{12}) \star \mu_{23}$ sends it to $\tau_{23}\big(\beta_{02}(p)\big)$. But both of those equal $\tau_{23}\big(\tau_{12}(\beta_{01}(p))\big)$, by Prop. 2.10.

Finally, $\gamma_{03}(p, -)$, computed either way, is the composite

$$\text{NT}\big(\tau_{23}\big(\tau_{12}(\beta_{01}(p))\big)\big) \xrightarrow{\gamma_{23}(\tau_{12}(\beta_{01}(p)),-)} \text{NT}\big(\tau_{12}(\beta_{01}(p))\big) \xrightarrow{\gamma_{12}(\beta_{01}(p),-)} \text{NT}(\beta_{01}(p)) \xrightarrow{\gamma_{01}(p,-)} \text{NT}(p)$$

□

Definition 2.12. Let CFG be the category whose objects are context-free grammars, with morphisms defined by Prop. 2.1 and composition defined by Prop. 2.8. The identity morphism on G is given by $(\mathrm{id}_N, \mathrm{id}_G, \mathrm{id}_{\mathrm{NT}(p)})$, i.e., identity maps.

We are now ready to assemble Prop. 2.3, Cor. 2.5, Prop. 2.10 and Prop. 2.11 into the main theorem of this paper. Intuitively, it says that tree is a functor from CFG to the category of sets. However, since we did not include a preferred start symbol in the data for context-free grammars (and much less did we assume that any such symbol would be preserved by morphisms), the target category is slightly more complicated. Let Mor(*Set*) be the category of maps of sets. An object of Mor(*Set*) is thus a function $f : X \to Y$ between arbitrary sets; a morphism from $f_1 : X_1 \to Y_1$ to $f_2 : X_2 \to Y_2$ consists of maps $u : X_1 \to X_2$ and $v : Y_1 \to Y_2$ such that

$$\begin{array}{ccc} X_1 & \xrightarrow{u} & X_2 \\ {\scriptstyle f_1}\downarrow & & \downarrow{\scriptstyle f_2} \\ Y_1 & \xrightarrow{v} & Y_2 \end{array}$$

commutes. Morphisms are composed 'horizontally'. Mor(*Set*) is an example of a diagram category (see e.g., Mac Lane 1978), but an alternative way to think of it is as the category of sets fibered over a base: $f : X \to Y$ can be thought of as the family of sets $f^{-1}(y)$ with $y \in Y$. Morphisms are then fiberwise maps.

Theorem 2.13. (a) tree *is a functor* CFG $\to$ Mor(*Set*). *It associates to a context-free grammar G the family of sets $\{\mathrm{tree}_G(x) \mid x \in N\}$. To a morphism of grammars $G_0 \to G_1$ it associates the map of families $\alpha : N_0 \to N_1$ and $\tau : \mathrm{tree}_{G_0}(x) \to \mathrm{tree}_{G_1}(\alpha(x))$, where $x \in N_0$.*

(b) *Let* UCFG *be the full subcategory of* CFG *whose objects are the unambiguous context-free grammars. $\hat{L}$ is a functor* UCFG $\to$ Mor(*Set*). *It associates to a context-free grammar G the family of sets $\{\hat{L}_G(x) \mid x \in N\}$. To a morphism of grammars $G_0 \to G_1$ it associates the map of families $\alpha : N_0 \to N_1$ and $f : \hat{L}_{G_0}(x) \to \hat{L}_{G_1}(\alpha(x))$ with $x \in N_0$, that sends $\mathsf{u} \in \hat{L}_{G_0}(x)$ to the sentential form generated by $\tau(T(\mathsf{u}))$, where $T(\mathsf{u})$ is the (unique) parse of u.*

(c) *L, sending G to the family $\{L_G(x) \mid x \in N\}$, is a subfunctor of $\hat{L}$.*

There exists a well-understood interplay between rational languages, finite state automata, and monoid objects in categories; the canonical reference is Arbib (1969). Category-theoretic properties of CFG (for example, the existence of pullbacks, filtered colimits or coproducts) as well as the roles that morphisms, functors, natural transformations etc. may play in formal language theory at higher levels of the Chomsky hierarchy, are much less explored.

3. Looking ahead

We have only dealt with two of the motivating examples. Neither of the other two can be described by a morphism $G \to G$ where G is any of the usual unambiguous context-free grammars for first order logic, or, I suspect, any context-free grammar for it. It should come as no surprise that there are limitations to the 'word processing power' of morphisms, as defined above. One expects that there exists a hierarchy of mappings between context-free grammars, just as there are hierarchies of languages, complexity classes, and so on. The goal of this final – much more speculative – section is to sketch further levels of this hierarchy. But first, here is one expression of the structural limitations of morphisms.

Proposition 3.1. *Suppose $(\alpha, \beta, \gamma) : G_0 \to G_1$ is a morphism of grammars with the property that for some constant K,*

$$\mathrm{depth}(\beta(p)) \leqslant K$$

for all $p \in G_0$. Then for all $x \in N_0$ and $T \in \mathrm{tree}_{G_0}(x)$,

$$\mathrm{depth}(\tau(T)) \leqslant K \cdot \mathrm{depth}(T)\ .$$

The proof is by induction on depth(T). Note that such a bound K always exists if G_0 is finite; however, our grammars (and alphabets) were not assumed to be so by default.

Example 3.2. Let L be the language of function terms for an associative binary operation (denoted by juxtaposition), fully parenthesized, with infinitely many variables available. The alphabet is

$$N = \{\ \mathsf{expr}\ \}$$
$$T = \{\ (\,)\, x_1\, x_2\ \ldots\ x_i\ \ldots\}$$

with unambiguous grammar

$$\mathsf{expr} \to x_1 \mid x_2 \mid \ \ldots\ \mid x_i \mid \ \ldots$$
$$\mathsf{expr} \to (\mathsf{expr}\,\mathsf{expr})$$

Let $\tau : L \to L$ be the mapping that sends an expression to its leftmost-parenthesized equivalent. For example,

$$((x_5x_3)((x_1x_3)x_2))$$

is to be sent to

$$((((x_5x_3)x_1)x_3)x_2)$$

If there was a morphism of grammars $(\alpha, \beta, \gamma) : G \to G$ inducing τ, it would have to satisfy

$$\beta(\mathsf{expr} \to x_i) = x_i$$

for all $i = 1, 2, \ldots$. Since there is only one other production in the grammar, namely,

$$\mathsf{expr} \ \to \ (\mathsf{expr}\,\mathsf{expr})$$

Prop. 3.1 would apply. However, for any positive integer d, let T be the term in variables $x_1, x_2, \ldots, x_{2^d}$ whose parse tree (ignoring parentheses) is the complete binary tree of depth d; e.g., for $d = 3$:

$$(((x_1x_2)(x_3x_4))((x_5x_6)(x_7x_8)))$$

$\tau(T)$ is a left-branching tree, with depth 2^d.

$$\Big\{\frac{\mathrm{depth}(\tau(T))}{\mathrm{depth}(T)} \ \mid \ T \in \mathrm{tree}_G(\mathsf{expr})\Big\}$$

is thus unbounded, and the mapping τ cannot correspond to any morphism of grammars.

This argument does not apply to our motivating example (c), replacement of free occurrences of a variable x in the input formula ϕ by some term t, since

$$\mathrm{depth}\big(\tau_{x\to\mathsf{t}}(\phi)\big) \leqslant \mathrm{depth}(\phi) + \mathrm{depth}(\mathsf{t})$$

always. (We have silently fixed an unambiguous context-free grammar G for first order logic.) However, no morphism $G \to G$ induces $\tau_{x\to\mathsf{t}}(\phi)$. The recursive rules

$$\tau_{x\to\mathsf{t}}(\phi \wedge \psi) \ \Rightarrow \ \tau_{x\to\mathsf{t}}(\phi) \wedge \tau_{x\to\mathsf{t}}(\psi)$$

$$\tau_{x\to\mathsf{t}}(\forall y\phi) \ \Rightarrow \ \forall y\tau_{x\to\mathsf{t}}(\phi)$$

showing that replacement descends the parse tree along boolean connectives and quantification with respect to variables other than x, conform perfectly to the combinatorial possibilities of a self-morphism of G. However, one has

$$\tau_{x \to \mathsf{t}}(\forall x \phi) \ \Rightarrow \ \forall x \phi \qquad (*)$$

since all free occurrences of x in ϕ become bound in $\forall x \phi$. $\tau_{x \to \mathsf{t}}(\forall x \phi)$ is thus not a function of $\tau_{x \to \mathsf{t}}(\phi)$, since ϕ cannot in general be reconstructed from $\tau_{x \to \mathsf{t}}(\phi)$. So $\tau_{x \to \mathsf{t}}$ cannot be computed by bottom-up induction, whereas translations induced by morphisms can always be.

Intuitively, a morphism of grammars applies the *same* functional transformation (itself!), iteratively, to subtrees of the input tree, whereas $(*)$ calls on a different transformation (namely, the identity) when the input has the form $\forall x \phi$. Recall that two functions $f, g : \mathbb{N} \to \mathbb{N}$ are defined by simultaneous recursion if $f(0)$ and $g(0)$ are given, and there exist functions F and G such that for $n > 0$,

$$f(n) = F\big(n, f(n-1), g(n-1)\big)$$
$$g(n) = G\big(n, f(n-1), g(n-1)\big) \ .$$

In the presence of a pairing function that codes the ordered pair $\langle f(n), g(n) \rangle$ as a single natural number, simultaneous recursion can be replaced by ordinary recursion. However, for tree transformations, simultaneous recursion on syntax has more expressive power than simple recursion.

Definition 3.3. Let G_0 and G_1 be context-free grammars in the alphabets N_0, T_0, N_1, T_1 as usual, and k a positive integer. A *k-morphism from G_0 to G_1 defined by simultaneous recursion* consists of the following data:

- mappings $\alpha_i : N_0 \to N_1$ for $i = 1, 2, \ldots, k$
- mappings β_i, for $i = 1, 2, \ldots, k$, assigning to each production $x \to \mathsf{s} \in G_0$ a parse tree from $\mathrm{tree}_{G_1}(\alpha_i(x))$
- for each $i = 1, 2, \ldots, k$ and each production $p \in G_0$, a function $\gamma_i(p, -)$ from $\mathrm{NT}(\beta_i(p))$ to $\mathrm{NT}(p)$ and a function $\delta_i(p, -)$ from $\mathrm{NT}(\beta_i(p))$ to $\{1, 2, \ldots, k\}$, with the property that for all $i = 1, 2, \ldots, k$ and all $t \in \mathrm{NT}(\beta_i(p))$, writing $j = \delta_i(p, t)$,

$$\alpha_j\big(\mathrm{label}(\gamma_i(p, t))\big) = \mathrm{label}(t) \ .$$

A k-morphism is, roughly, a k-tuple of grammatical transformations that are intertwined via the function δ: the i-th transformation can call on the j-th transformation to act on a subtree of the input tree. The maps α_i provide the initial values. There is no circular dependency, since each recursive call applies to a lower-level *subtree* of the input tree. More precisely,

Proposition 3.4. *A k-morphism of grammars from G_0 to G_1 induces, for each $i = 1, 2, \ldots, k$ and $x \in N_0$, a mapping*

$$\tau_i : \text{tree}_{G_0}(x) \to \text{tree}_{G_1}(\alpha_i(x)) \ .$$

Proof. For $T \in \text{tree}_{G_0}(x)$, define the $\tau_i(T) \in \text{tree}_{G_1}(\alpha_i(x))$ simultaneously by induction on the depth of T:

- If $\text{depth}(T) = 0$, then T must be x itself, and $\tau_i(T)$ is defined to be $\alpha_i(x)$.
- If $\text{depth}(T) > 0$, let $x \to \mathsf{s} \in G_0$ be the top production in T. Write p for $x \to \mathsf{s}$ for brevity. As usual, $\text{NT}(p)$ can be identified with a subset of s, the locations of the non-terminal symbols in s. Since G_0 is context-free, each $s \in \text{NT}(p)$ induces a subtree T_s of T with s as root. For each $i = 1, 2, \ldots, k$ and $t \in \text{NT}(\beta_i(p))$, writing $j = \delta_i(p, t)$, graft the tree $\tau_j(T_{\gamma_i(p,t)})$ on $\beta_i(p)$ with t as root. $\tau_i(T)$ is defined to be the resulting tree.

Since $\text{depth}(T_s) < \text{depth}(T)$ for all $s \in \text{NT}(p)$, $\tau_j(T_s)$ is defined by the induction hypothesis. Note that $\tau_j(T_s)$ belongs to $\text{tree}_{G_1}(\alpha_j(\text{label}(s))$ by the induction assumption, and $\alpha_j\big(\text{label}(\gamma_i(p, t))\big) = \text{label}(t)$ by Def. 3.3. That is, the non-terminal symbol at the root of $\tau_j(T_{\gamma_i(p,t)})$ coincides with the non-terminal symbol at the location t. Since G_1 is a context-free grammar, the graft is well-defined, and $\tau_i(T)$ will belong to $\text{tree}_{G_1}(\alpha_i(x))$ as desired. ☐

When finding $\tau_i(T)$ by recursion from root to leaves on T, one can restrict to computing $\tau_j(T_s)$ only for those subtrees T_s of T and values $j \in \{1, 2, \ldots, k\}$ that are called for by the indexing function δ. When using bottom-up induction, the entire k-tuple of values $\big(\tau_1(-), \tau_2(-), \ldots, \tau_k(-)\big)$ needs to be computed for all subtrees of T.

Mutatis mutandis, the results of the previous section, from Prop. 2.3 to Prop. 3.1, remain valid for morphisms defined by simultaneous recursion. The composition of a k-morphism from G_0 to G_1 and n-morphism from G_1 to G_2 will be a $k \cdot n$-morphism from G_0 to G_2. Composition is associative, and tree_G becomes a functor from CFG to tuples of functions of sets. The details, while not conceptually complicated, are quite tedious (largely for notational reasons) and will not be needed here.

The reader is invited to define the pair of transformations $(\tau_{x\to\mathtt{t}}, \text{id})$ by simultaneous recursion on the syntax of first order logic. $\tau_{x\to\mathtt{t}}$ calls itself and id, while the identity transformation calls itself only. The fact that the treatment of descendant nodes is inherited from their parent nodes is reminiscent of attribute grammar.

Note that $\phi_{x\to\mathtt{t}}$, replacing all free occurrences of the variable x in the formula ϕ by the term $\mathtt{t}$, is the least complicated of the multitude of operations involving variable replacement and binding. If a free variable in $\mathtt{t}$ is captured by a quantifier in ϕ, then ϕ will no longer imply its instance $\phi_{x\to\mathtt{t}}$; to preserve the intended logical meaning, the dummy variable appearing in the capturing quantifier in ϕ should be renamed first, to a variable not occurring in ϕ or $\mathtt{t}$. However, the function that returns a variable *not* occurring in a given formula does not have a canonical value, and is not easily describable in terms of language operations. A related, and much researched, issue is the formalization of *explicit substitution* in lambda calculi (Abadi et al. 1990): under explicit substitution, the operation $x \to \mathtt{t}$ does not belong to the meta-language, but is part of the language itself. On the other hand, there seem to exist few studies, from the viewpoint of mathematical linguistics, of the syntax of substitutions through de Bruijn indices or Bourbaki's variable-free notation (Mathias 1999).

Prop. 3.1 does not apply either to our fourth (and last) motivating example, transforming first order formulas ϕ to negation normal form $\textsc{nnf}(\phi)$, since $\text{depth}(\textsc{nnf}(\phi)) \leqslant \text{depth}(\phi)$ always. But $\textsc{nnf}$ cannot be induced by a morphism, or in fact k-morphism. The standard context-free grammars of first order logic contain the production

$$\mathsf{expr} \to \neg\,\mathsf{expr}$$

But $\beta(\mathsf{expr} \to \neg\,\mathsf{expr})$ cannot contain any terminal symbols; any nonterminal other than 'expr'; or more than one copy of 'expr': each of those possibilities would be inconsistent with the fact that $\textsc{nnf}(\neg\neg\phi) = \textsc{nnf}(\phi)$. So $\beta(\mathsf{expr} \to \neg\,\mathsf{expr})$ is forced to be 'expr', which of course is incompatible with the negation normal form of $\neg\phi$ for atomic ϕ.

Intuitively, the issue is that the rewrite rules

$$\begin{aligned} \neg\neg\phi &\Rightarrow \phi \\ \neg(\phi \wedge \psi) &\Rightarrow \neg\phi \vee \neg\psi \\ \neg(\phi \vee \psi) &\Rightarrow \neg\phi \wedge \neg\psi \\ \neg\,\forall x\phi &\Rightarrow \exists x \neg\psi \\ \neg\,\exists x\phi &\Rightarrow \forall x \neg\psi \end{aligned}$$

may introduce new instances of the negation symbol on their right hand sides. It requires a moment of thought to verify that this set of rules is *noetherian* (starting with any formula, they cannot be applied infinitely often) and many more moments of thought to verify that they are *confluent* (every formula has a unique negation normal form, even if the above rewrite rules are applied to arbitrary *subformulas* first, in any order, until no rule applies anywhere).

Recall that a *term rewrite system* (TRS) is an unordered set of rewrite rules acting on function terms in some fixed signature. A TRS is called convergent if it is both noetherian and confluent (Baader & Nipkow 1998). All four motivating examples, and Example 3.2 as well, belong to the family of convergent TRS, adapted from the unambiguous grammar of function terms to the general setting of context-free grammars. The fact that the effect of morphisms on parse trees can be computed by both bottom-up and top-down recursion, as well as Lemma 2.9, can be seen as corollaries of confluence.

It is quite challenging, however, to fashion a category out of convergent TRS. To begin with, neither the confluence nor the noetherianness of TRS is, in general, decidable (though, curiously, the confluence of noetherian TRS *is* decidable). Secondly, a famous example due to Toyama shows that the disjoint union of two convergent TRS need not be convergent. Thus the composite of two TRS cannot, in general, be defined as the disjoint union of their underlying rules. There exist, however, sufficient conditions for the modularity of convergence for TRS. Alternatively, one can experiment with ordered (prioritized) rewriting rules.

In a different direction, the notion of morphism of context-free grammars could be broadened to allow for non-determinism: several right-hand sides of the component β. Finally, the focus on parse trees is, to some extent, restrictive: the domain of these transformations could be any set of node-labeled rooted trees closed under taking subtrees.

I hope to elaborate some of these ideas in later publications. In closing, let me return to the quote from Chomsky that opened this article. *Suppose* that the only gift linguistics ever gave mathematics was, indeed, the notion of context-free grammar. Let's play with this present: expand the focus from context-free grammars to maps of context-free grammars (from objects to morphisms) and I think we will agree that linguistics has given mathematics a gift that keeps on giving.

References

Abadi, M., L. Cardelli, P.-L. Curien and J.-J. Lévy. 1990. Explicit substitutions. Digital Systems Research Center Technical Report SRC-RR-54.

Aho, A. V. and J. D. Ullman. 1972. The theory of parsing, translation, and compiling. Volume 1: Parsing. Englewood Cliffs, NJ: Prentice-Hall.

Arbib, M. A. 1969. Theories of abstract automata. Englewood Cliffs, NJ: Prentice-Hall.

Atallah, M. J. and M. Blanton (eds.). 2010. Algorithms and theory of computation handbook. Volume 2: Special topics and techniques. Boca Raton, FL: Chapman & Hall/CRC.

Baader, F. and T. Nipkow. 1998. Term rewriting and all that. Cambridge: Cambridge University Press.

Chomsky, N. 1982. The generative enterprise. A discussion with Riny Huybregts and Henk van Riemsdijk. Dordrecht: Foris.

Chomsky, N. 2004. The generative enterprise revisited: Discussions with Riny Huybregts, Henk Van Riemsdijk, Naoki Fukui and Mihoko Zushi. With a new foreword by Noam Chomsky. Berlin & New York: Mouton de Gruyter.

Mac Lane, S. 1978. Categories for the working mathematician (Second edition). New York: Springer.

Mathias, A. R. D. 1999. A term of length 4,523,659,424,929. Preprint. www.dpmms.cam.ac.uk/~ardm/inefff.pdf.

Mendelson, E. 2010. Introduction to mathematical logic (Fifth edition). Boca Raton, FL: Chapman & Hall/CRC.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 105–140

OT grammars don't count, but make errors

TAMÁS BIRÓ
ELTE Eötvös Loránd University, Budapest
tamas.biro@btk.elte.hu

KEYWORDS

Optimality Theory
Harmonic Grammar
strict domination
simulated annealing
ICS Architecture

ABSTRACT

Our goal is to compare *Optimality Theory* (OT) to *Harmonic Grammar* (HG) with respect to simulated annealing, a heuristic optimization algorithm. First, a few notes on Smolensky's ICS Architecture will bridge the gap between connectionist HG and symbolic HG. Subsequently, the latter is connected to OT via q-HG grammars, in which constraint C_i has weight q^i. We prove that q-HG converges to OT if $q \to +\infty$, even if constraint violations have no upper bound. This limit shall be referred to as the *strict domination limit*. Finally we argue that q-HG in the strict domination limit shares with OT a remarkable feature: simulated annealing does not always converge to 100% precision, even if the algorithm is offered ample time. Globally non-optimal local optima produced at slow pace will be viewed as *irregular forms*.

There were three greengrocers in Sziget street. Kardos, the first owner, put a sign in his window: "Best vegetables in town!". Then Kerekes, the owner of the second shop, raised the bid by posting: "Best vegetables of the world!" The third owner, Kohn, had a hard time. What should he do now? He finally decided to write on his door: "Best vegetables of the street!"

"Errare humanum est", said the hedgehog when he climbed down the wire brush.

1. Do grammars count?

Well, we all are linguists, and so grammars do matter to us. And yet, it has become common wisdom in our profession that "grammars don't count". To quote Kornai (2008, 250): "the heavy emphasis on noncounting languages originates in an apocryphal remark of John von Neumann: 'The brain does not use the language of mathematics'".

This maxim has been used in several ways. Whether counting is legitimate or mistaken in language description, whether languages count segments, syllables or words (e.g., among many others, McCarthy 2002; González 2005; Watanabe 2009; Graf 2017), has been a long debate that shall not be our concern here. Our question is whether language *models* should make use of counting. More precisely: what is the consequence of the fact that *Optimality Theory* (Prince & Smolensky 1993; 2004) avoids counting, whereas *Harmonic Grammar* (Smolensky & Legendre 2006) does count?

Even within Optimality Theory, "to count or not to count" is a question raised multiple times. Apropos constraint violations, we all know that the violation level $C_k(x)$ assigned by constraint C_k to candidate x is usually a number of "stars". Yet, some constraints are simply categorical, binary, with range $\{0, 1\}$ or $\{\texttt{true}, \texttt{false}\}$, $\{\texttt{satisfies}, \texttt{violates}\}$: the last syllable of a word is either parsed into a foot or it is not, *Wh*-movement either has taken place or has not, the meaning is either faithfully expressed in the form or it is not, and so forth. Many other constraints are binary within some "locus", but the candidate itself is composed of several such "loci", and so they can be violated multiple times. These constraints *count* the number of marked segments or disfavored foot types or unfaithful features in the candidate. Some other constraints again may be violated to several degrees: for instance, the larger the distance of the head foot to some word edge (measured as the number of intervening syllables), the graver the violation of this alignment constraint by the candidate. Finally, some constraints can be gradually violated by several loci, and a non-trivial axiom of OT is that these constraints simply sum up the violations by the loci. Which of these constraints should and which should not be used is again a long story (McCarthy 2002; 2003; Bíró 2003; Eisner 1997).

In standard OT, the counting by a constraint C_k is usually only a technicality, which boils down to the question which of $C_k(x)$ and $C_k(y)$ is *greater* (a more severe case of constraint violation). The specific numerical values actually only matter in Harmonic Grammar. Thus we arrive at the question that shall concern us here: is counting involved when constraints are combined into a single architecture?

Thus, given is a set $\{C_1, C_2, \ldots, C_n\}$ of constraints. Of these constraints, both *Harmonic Grammar* (HG) and *Optimality Theory* (OT) build up an objective function (target function) $H(x)$ to be optimized. While HG uses a weighted sum of the violations $C_k(x)$ (refer to equation (6) later), OT creates a vector, best known as the row corresponding to candidate x in an OT tableau (cf. (9)). Both approaches postulate the output (e.g., surface form) SF(u) corresponding to input (e.g., underlying form) u to be the most harmonic element of the candidate set Gen(u):

$$\mathrm{SF}(u) = \underset{x \in \mathrm{Gen}(u)}{\arg\mathrm{opt}}\ H(x) \tag{1}$$

In Harmonic Grammar, optimization is simply minimization in terms of the arithmetic *greater than* relation. Whereas in OT, it is the *lexicographic order* on a set of real-valued vectors: you compare the first components of the two vectors (the violations of the highest ranked constraint); if they are equal, then you proceed with comparing their second components; and so forth (Eisner 2000; Jäger 2002; Prince 2002).

The output SF(u) can be conceived of as the *grammatical form*, that is, the form predicted by the grammar, the model of human *linguistic competence* (Newmeyer 1983). The next step is to find the candidate x that optimizes the objective function $H(x)$, a procedure that has been compared to *linguistic performance* (Smolensky & Legendre 2006; Biró 2006).

What procedure shall we use to find the optimal candidate? Similarly to the third greengrocer in Sziget street, we shall optimize locally. But as the hedgehog warns us, local optimization can go wrong. We discuss simulated annealing, a probabilistic hill climbing algorithm that performs local search, comparing its behavior with OT to its behavior with HG.

This article is structured as follows. Section 2 presents the link between connectionist Harmonic Grammar and symbolic Harmonic Grammar, also summarizing Smolensky's ICS Cognitive Architecture in passing. Subsequently, section 3 stretches the connection to Optimality Theory by introducing the concept of q-HG, a variant of Harmonic Grammar with exponential weights. As a new mathematical result, we show that q-HG converges to OT as the base of the exponents q grows infinite, even if no upper bound exists on the number $C_k(x)$ of violation marks. Then, section 4 introduces simulated annealing, before section 5 elaborates on why it works in most cases. In contrast to that, section 6 explains the main message of this paper: simulated annealing can fail in the *strict domination limit* ($q \to +\infty$). This point is illustrated by computer experiments in section 7, before drawing the conclusions in section 8.

2. From connectionist HG to symbolic HG

In order to understand why the optimization technique called *simulated annealing* is relevant for Optimality Theory, let us first recapitulate the connectionist idea behind OT. This section can also be read as an intro-

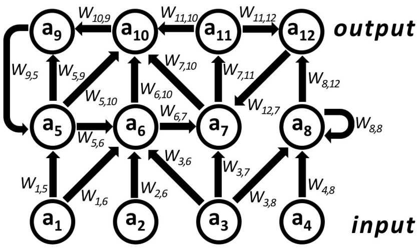


Figure 1: A Boltzmann machine with twelve nodes, an input layer and an output layer

duction to Paul Smolensky's *Integrated Connectionist/Symbolic Cognitive Architecture* (ICS) (Smolensky & Legendre 2006).

A *Boltzmann machine* (Fig. 1) is a set of N nodes $\left(a_i\right)_{i=1}^{N}$, each with activation value a_i, a real number. It also includes, for each i and j, the connection strength W_{ij} of the arc from node i to node j, a real number again. Skipping some technical details often included in the literature on Boltzmann machines, let the *energy* E of the Boltzmann machine – the negative of the Harmony mentioned earlier and to be introduced soon – be simply the following sum of multiplications, over the connections i to j:

$$E = \sum_{i,j=1}^{N} a_i \cdot W_{ij} \cdot a_j. \tag{2}$$

In *connectionist HG*, a *constraint* C_k is a set of partial connection strengths: some W_{ij}^k for each arc (i, j) in the network (Fig. 2). If the weight of C_k is w_k, and there are n constraints, then the total connection strength for arc (i, j) in the ensuing network is

$$W_{ij} = \sum_{k=1}^{n} w_k \cdot W_{ij}^k. \tag{3}$$

The notion of *candidate* in symbolic OT and HG, introduced in the previous section, corresponds to an activation pattern in connectionist HG. Some of the nodes describe the input (e.g., underlying representation), and are *clamped* (set) during computation. Some other nodes encode the output by the end of the computation. The rest of the nodes are "hidden". They may correspond to hidden (or covert) information, not present either in the input or in the output, but encoded in the candidate, as it plays some role in linguistic theory: syllable structure (Soderstrom et al. 2006),

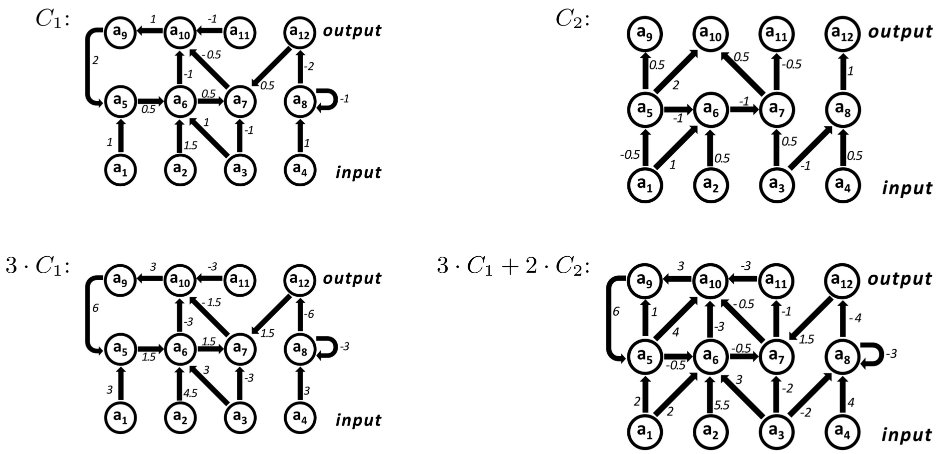


Figure 2: Examples of two constraints as partial connection strengths (upper row), and their weighted sums (linear combinations) (lower row). A missing arc means strength 0.

prosodic and syntactic parsing brackets, correspondence relations between input and output (cf. McCarthy & Prince 1995), intermediate levels of representations (e.g., Boersma 2011), and so forth.

During computation, the Boltzmann machine moves toward a (locally) minimal energy state, with its input nodes clamped and its output nodes eventually encoding the optimal output. The *candidate set* Gen(u) for input u is thus formed by the possible states of the network, with its input nodes fixed to encode u.

From (2) and (3), we get the *energy of a connectionist HG model* in state $A = \left(a_i\right)_{i=1}^{N}$:

$$E[A] = \sum_{i,j=1}^{N} a_i \cdot W_{ij} \cdot a_j = \sum_{i,j=1}^{N} a_i \cdot \sum_{k=1}^{n} w_k \cdot W_{ij}^{k} \cdot a_j = \sum_{k=1}^{n} w_k \cdot \sum_{i,j=1}^{N} a_i \cdot W_{ij}^{k} \cdot a_j. \quad (4)$$

We may now identify $C_k[A] = \sum_{i,j=1}^{N} a_i \cdot W_{ij}^{k} \cdot a_j$ as the *violation of constraint* C_k by the activation pattern (i.e., candidate) $A = \left(a_i\right)_{i=1}^{N}$. In turn,

$$E[A] = \sum_{k=1}^{n} w_k \cdot C_k[A]. \quad (5)$$

This is the equation that creates the bridge between connectionist HG and symbolic HG.

For instance, the violation of constraint C_1 in Fig. 2 is $a_1 \cdot a_5 + 0.5 \cdot a_5 \cdot a_6 + 1.5 \cdot a_2 \cdot a_6 + \ldots$ Similarly, the violation of C_2 turns to be $-0.5 \cdot a_1 \cdot a_5 - a_5 \cdot a_6 + a_1 \cdot a_6 + 0.5 \cdot a_2 \cdot a_6 + \ldots$ As a result, a connectionist harmonic grammar with weights $w_1 = 3$ and $w_2 = 2$ corresponds to the linear combination $2 \cdot a_1 \cdot a_5 - 0.5 \cdot a_5 \cdot a_6 + 2 \cdot a_1 \cdot a_6 + 5.5 \cdot a_2 \cdot a_6 + \ldots$

To summarize, a candidate x in symbolic HG corresponds to an activation pattern A of the Boltzmann network. A constraint C_k is a set of partial connection strengths W_{ij}^k, and its violation $C_k(x)$ by candidate x turns into the sum $\sum_{i,j=1}^{N} a_i \cdot W_{ij}^k \cdot a_j$. Hereby, the energy $E[A]$ of the Boltzmann network will map to the (negative) Harmony $H(x)$ of the HG grammar. Boltzmann machines, as a connectionist technique, minimize their energy, and so symbolic HG linguistic models also ought to optimize their harmony.

3. From symbolic HG to symbolic OT, via *q*-HG

To summarize, we have derived the connection between minimizing the energy $E[A]$ in connectionist HG and maximizing the harmony $H(x)$ in symbolic HG. The opposite directions of optimization can be taken care of with a negative sign, and it has historical reasons. To tell the truth, I personally prefer the minimization perspective even in OT, since the best candidate is the one that violates the constraints the least. We will nevertheless have to introduce that negative sign in order to maintain the view according to which the best candidate maximizes its harmony, given constraint weights w_k:

$$H(x) = -\sum_{k=1}^{n} w_k \cdot C_k(x) \tag{6}$$

Now we proceed further towards *Optimality Theory*. It has become customary, especially in the literature on learning, to assign a real-valued *rank* r_k to each constraint C_k (Boersma 1997; Boersma & Hayes 2001). The higher its rank, the higher the constraint will be ranked in OT. The most direct connection between *weights* and *ranks* is identifying them: $w_k = r_k$ ("linear HG"). A less self-evident connection, *exponential HG* (Boersma & Pater 2008), has however been more frequently employed: $w_k = \exp(r_k)$, with some base larger than 1 (such as 2 or 10 or $e = 2.71\ldots$). Exponentiating the ranks has the advantage that learning will never produce negative weights (Pater 2009), as well as that it also makes learning more efficient (cf. the inefficiency of learning linear HG, as demonstrated by Magri 2016).

Earlier, I have introduced an approach called *q-Harmonic Grammar* in which $w_k = q^{r_k}$ (Biró 2009). Having the value of $q > 1$ fixed, we may define a 2-HG grammar, a 10-HG grammar or a 1.23-HG grammar, if $q = 2$ or 10 or 1.23, respectively. But we can also change the value of q. The difference between exponential HG and q-HG is a question of perspective: the former sets the base of exponentiation, and considers it merely as a technical detail, whereas the latter views it as an interesting tunable parameter.

On the one hand, a change of the basis q from q_1 to q_2 is technically equivalent to multiplying all ranks by the factor $\frac{\log q_2}{\log q_1}$. On the other hand, if the ranks are kept fixed, then increasing q is how we can get HG to turn into OT: we are demonstrating momentarily that under certain conditions an OT grammar and a q-HG grammar with the same constraint ranks define the same language in the $q \to +\infty$ limit.

Parameter q becoming infinitely large will be called the *strict domination limit*. The motivation of the expression is that the key difference between HG and OT is *strict domination*: if the two approaches predict different language typologies, then it is because HG, but not standard OT, allows *counting cumulativity* and *ganging-up cumulativity* (Jäger & Rosenbach 2006).

Table 1: Counting cumulativity: [y] is more harmonic than [x] if $q = 3$

/u/	C_2	C_1	3-HG	5-HG	OT
	$r_2 = 2$	$r_1 = 1$			
$q = 3$	$w_2 = 3^2 = 9$	$w_1 = 3^1 = 3$			
$q = 5$	$w_2 = 5^2 = 25$	$w_1 = 5^1 = 5$			
[x]		****	-12	☞ -20	☞
[y]	*		☞ -9	-25	

Table 2: Ganging-up cumulativity: [y] is more harmonic than [x] if $q = 3$

/u/	C_3	C_2	C_1	3-HG	5-HG	OT
	$r_3 = 3$	$r_2 = 2$	$r_1 = 1$			
$q = 3$	$w_3 = 27$	$w_2 = 9$	$w_1 = 3$			
$q = 5$	$w_3 = 125$	$w_2 = 25$	$w_1 = 5$			
[x]		**	****	-30	☞ -70	☞
[y]	*			☞ -27	-125	

Tableaux 1 and 2 illustrate the point. The best candidates, shown by the pointing hand, are calculated with respect to the hierarchy $(C_3 \gg) C_2 \gg$

C_1 in OT; and as the weighted sum of the violations in q-HG, the weights being $w_i = q^{r_i}$. In both tableaux, candidate [x] is more harmonic than [y] for OT and 5-HG. However, for 3-HG, tableau 1 shows how multiple violations of the lower ranked constraint C_1 can turn the candidate [y] more harmonic. Similarly, in tableau 2, the lower ranked two constraints, C_2 and C_3, gang up: while neither of them alone could make [x] worse than [y], taking them together results in [y] winning over [x].

In both cases, 5-HG behaves like OT, and any q-HG would do so if $q \geq 5$. It has been long known (Prince & Smolensky 2004, 236) that a sufficient criterion for a harmonic grammar with an exponential weight system to display OT-like behavior is that the base of the exponential weights be not less than the highest amount of stars in a cell (which is 4 in our example) plus 1. This is why 5-HG is equivalent to OT, but not 3-HG. (For a reversed approach, refer to Prince 2002.)

Let us now formalize this observation. $\mathcal{U}$ shall be the set of underlying forms – the domain of the universal Gen function – which is postulated to be universal by the *Richness of the Base* principle (Prince & Smolensky 2004, 225). Moreover, let us posit that our n constraints take non-negative integer values: for $k = 1, \ldots, n$, the constraint C_k is a mapping from $\bigcup_{u \in \mathcal{U}} \text{Gen}(u)$ to $\mathbb{N}_0$. This last requirement will play a crucial role in the proof to be presented. While it certainly applies to most linguistic models in the OT and HG literature, it poses some limitations to the generalizability of the framework.

Without loss of generality, we can assume that the indices of the constraints reflect their ranking. Consequently, our OT grammar shall be

$$C_n \gg C_{n-1} \gg \ldots \gg C_1 \tag{7}$$

This OT grammar can be matched to the q-HG grammar with $r_k = k$ and $w_k = q^k$ (remember that $q > 1$). The point of interest is whether these two grammars generate the same language. Put it differently, the following two Harmony functions are compared:

$$H_q(x) = -\sum_{k=1}^{n} q^k \cdot C_k(x) \tag{8}$$

$$H_{\text{OT}}(x) = \Big(-C_n(x), -C_{n-1}(x), \ldots, -C_1(x)\Big) \tag{9}$$

Equation (1), reformulated here, defines a grammar for either kind of Harmony functions:

$$\text{SF}(u) = \underset{x \in \text{Gen}(u)}{\arg\max}\; H(x) \tag{10}$$

(for all $u \in \mathcal{U}$). Such a grammar maps an underlying form u to a surface form s, if and only if $H(s) \succeq H(x)$ for all $x \in \text{Gen}(u)$. $\text{SF}(u)$ is the set of these optimal candidates. In the case of q-HG, the values of H_q are compared using the arithmetic *greater than or equal to* relation $\geq$, whereas in OT, the *lexicographic order* $\succeq_{\text{lex}}$ compares the H_{OT} vectors. In the former case, the set of optimal candidates will be denoted as $\text{SF}_q(u)$, and in the latter case, as $\text{SF}_{\text{OT}}(u)$.

We now prove a theorem that guarantees that the OT grammar (7) and the corresponding q-HG grammar (8) map any $u \in \mathcal{U}$ to the same surface form(s), if q is sufficiently large. This fact has been long known (Prince & Smolensky 2004, 236), but only if the number of violations admitted by the constraints were limited. We now show that no such upper limit is required, if the constraints take integer values.

Theorem 1. *Given are non-negative integer constraints $C_n, C_{n-1}, \ldots, C_1$ (ordered by their indices) and a Generator function* Gen. *Then, for any underlying form $u \in \mathcal{U}$ there exists some threshold $q_0 \geq 1$ such that for all $q > q_0$, $\text{SF}_{OT}(u) = \text{SF}_q(u)$.*

Proof. For any given $u \in \mathcal{U}$, we shall construct such a q_0. In this proof, the symbols s, s_1, s_2 and x will always denote elements of $\text{Gen}(u)$.

First, observe that if $s_1 \in \text{SF}_{\text{OT}}(u)$ and $s_2 \in \text{SF}_{\text{OT}}(u)$, then from the definition of the optimal set $\text{SF}_{\text{OT}}(u)$, we obtain $H_{\text{OT}}(s_1) \succeq_{\text{lex}} H_{\text{OT}}(s_2)$ and $H_{\text{OT}}(s_2) \succeq_{\text{lex}} H_{\text{OT}}(s_1)$; from which it follows that they share the same violation profile. That is, they violate each constraint to the same level: $C_k(s_1) = C_k(s_2)$ for all k.

In turn, it is well-founded to introduce the threshold q_0 as

$$q_0 = 1 + \max\left\{C_k(s), C_{k-1}(s), \ldots, C_1(s)\right\}$$

for whichever $s \in \text{SF}_{\text{OT}}(u)$. Since the constraints are postulated to have a non-negative range, $q_0 \geq 1$ follows. Now we have to show $\text{SF}_{\text{OT}}(u) = \text{SF}_q(u)$ to hold for all $q > q_0$.

If $s_1 \in \text{SF}_{\text{OT}}(u)$ and $s_2 \in \text{SF}_{\text{OT}}(u)$, then they violate each constraint to the same level, and so $H_q(s_1) = H_q(s_2)$, for any q. In order to complete our proof, it remains to be shown that if $q > q_0$, $s \in \text{SF}_{\text{OT}}(u)$ and $x \notin \text{SF}_{\text{OT}}(u)$, then $H_q(s) > H_q(x)$. Candidates that are suboptimal for H_{OT} are also suboptimal for H_q.

Since $s \in \text{SF}_{\text{OT}}(u)$ and $x \notin \text{SF}_{\text{OT}}(u)$, the vector $H_{\text{OT}}(s)$ is strictly lexicographically greater than the vector $H_{\text{OT}}(x)$. This means that there exists some "fatal constraint" C_f such that for all $k > f$, $C_k(s) = C_k(x)$, and $-C_f(s) > -C_f(x)$. Since our constraints take integer values, we conclude that $C_f(s) - C_f(x) \leq -1$.

Moreover, observe that for any k, $C_k(s)-C_k(x) < q-1$. This inequality holds because by the above definition of q_0, $C_k(s) \leq q_0-1 < q-1$, whereas by the non-negativity of all constraints, $C_k(x) \geq 0$.

These two inequalities on the differences of the violations yield, for all $q > q_0 \geq 1$,

$$\begin{aligned}
H_q(x) - H_q(s) &= \sum_{k=1}^{n} \left[C_k(s) - C_k(x)\right] \cdot q^k = \\
&= \left[C_f(s) - C_f(x)\right] \cdot q^f + \sum_{k=1}^{f-1} \left[C_k(s) - C_k(x)\right] \cdot q^k < \\
&< -1 \cdot q^f + \sum_{k=1}^{f-1} (q-1) \cdot q^k = -q^f + (q-1) \cdot \sum_{k=1}^{f-1} q^k = \\
&= -q^f + (q-1) \cdot \frac{q^f - q}{q-1} = -q < 0.
\end{aligned}$$

That is, $H_q(s) > H_q(x)$ indeed holds. To summarize, for each $u \in \mathcal{U}$, we have proposed a $q_0 \geq 1$ such that for all $q > q_0$, the elements of $\mathrm{SF}_{\mathrm{OT}}(u)$ are equally harmonic in q-HG; but they are more harmonic with respect to H_q than the candidates *not* in $\mathrm{SF}_{\mathrm{OT}}(u)$. Thus, OT and q-HG map u to the same optimal subset $\mathrm{SF}_q(u) = \mathrm{SF}_{\mathrm{OT}}(u) \subseteq \mathrm{Gen}(u)$. □

Obviously, nothing requires that a single q_0 work for all elements of $\mathcal{U}$; rather q_0 is dependent on u. But as q grows, more and more underlying forms will be mapped to the same surface forms by OT and by q-HG. Let $q_{0(u)}$ be some threshold q_0 for u, such as the one constructed in the proof of Theorem 1. Since $\mathcal{U}$ is most often a countable set, we can sort its elements by $q_{0(u)}$. Let the $q_{0(u)}$ value of the kth element of $\mathcal{U}$ in this list be $q_{0[k]}$. Now, if you wish your q-HG grammar to map at least k elements of $\mathcal{U}$ to the same output as the corresponding OT grammar does, then you should have $q > q_{0[k]}$.

As an example, remember tableaux 1 and 2. We have seen that $q_0 = 5$ is a good threshold: for all $q > q_0 = 5$, the q-HG grammar corresponding to the OT grammar will yield the output that is also most harmonic in the OT approach. But imagine now a different input, $/u'/$, whose OT winner $[x']$ incurs 6 violations by constraint C_1. This second input will require $q_{0(u')} = 7$, a higher threshold. And yet, you can set q to 7.1, and your q-HG grammar turns equivalent to OT for both inputs. And so forth. Even if you do not have an *a priori* upper bound of the number of stars assigned by C_1, and even if you do not want to restrict the input set arbitrarily, you will know: whenever you are about to compute the most harmonic element of a candidate set, you can have a value of q such that q-HG may be used instead of OT.

If a *language* is the way it maps inputs (underlying forms) onto outputs (surface forms), then the functions (set of mappings) $\mathrm{SF}_{\mathrm{OT}}$ and SF_q are simply the languages generated by an OT grammar and by a q-HG grammar, respectively. Alternatively, the Chomskyan *E-languages* would be the ranges of $\mathrm{SF}_{\mathrm{OT}}$ and of SF_q, respectively.

The theorem just proven can be reformulated as follows: the language generated by q-HG converges to the language generated by OT, as q grows infinitely large; that is,

Corollary 2.

$$\lim_{q \to +\infty} \mathrm{SF}_q = \mathrm{SF}_{OT} \quad \text{pointwise.}$$

Here the *pointwise convergence* of a sequence of functions on $\mathcal{U}$ is understood as follows: for any $u \in \mathcal{U}$ there exists some q_0 such that for all $q > q_0$, $\mathrm{SF}_q(u) = \mathrm{SF}_{\mathrm{OT}}(u)$. The limit $q \to +\infty$ has been called the *strict domination limit* (Biró 2009).

Before proceeding, a remark is in order. The proof crucially relied on the constraints taking non-negative integer values. In the general case, however, Corollary 2 might still hold, even if in a weaker sense.

Take the following HG and OT grammars: candidates are non-negative real numbers ($\mathrm{Gen}(u) = \mathbb{R}_0^+$), while the two constraints are $C_2(x) = (x-1)^2$ and $C_1(x) = x$. In OT, the single best candidate for the highest ranked constraint C_2 is $x^*_{\mathrm{OT}} = 1$. All other candidates incur more violations by C_2, and so C_1 plays no role. In q-HG, however, $H_q(x) = -q^2 \cdot (x-1)^2 - q \cdot x$, which takes its maximum at $x^*_q = \frac{2q-1}{2q}$. For no real q will $x^*_q = x^*_{\mathrm{OT}}$; and so no q_0 exists such that $\mathrm{SF}_q(u) = \mathrm{SF}_{\mathrm{OT}}(u)$ for all $q > q_0$.

Observe, though, that $\lim_{q \to +\infty} x^*_q = x^*_{\mathrm{OT}}$. In a weaker sense, Corollary 2 still holds, at least for this specific example: for all $u \in \mathcal{U}$ and all $\epsilon > 0$, there exists some q_0 such that for all $q > q_0$, the distance of $\mathrm{SF}_q(u)$ and $\mathrm{SF}_{\mathrm{OT}}(u)$ is less than ϵ. Readers worried about the linguistic relevance of this example should note that the factorial typology includes candidates 1 and 0, and so it can be seen as a model of how a continuous phonetic feature maps to categorical phonology: it is either present or absent from a language. And yet, for candidates that are symbols or objects without a meaningful distance metric, it would be hard to formulate a similar conjecture of convergence.

4. Simulated annealing for symbolic Harmonic Grammars

Once we have defined how our grammars map an input (or underlying form) onto an output (or surface form) as an optimum defined by eq. (1) or eq. (10), in the second half of this paper we turn to the next question: how to find this optimum? The Boltzmann machines underlying connectionist HG immediately come with an answer: *simulated annealing*.

In the case of symbolic OT, the answer may be much less obvious, and even 'hard'. While most of our colleagues happily rely on their intuitions, Lauri Karttunen (2006) demonstrated "the insufficiency of paper-and-pencil linguistics", arguing for finite-state implementations of OT. Finite-state OT, however, imposes requirements that are met by many, but not all linguistic models (cf. e.g., Eisner 1997, Jäger 2002 and Bíró 2003, and references therein). Further approaches include dynamic programing (or chart parsing; Tesar & Smolensky 2000) and genetic algorithms (Turkel 1994; Pulleyblank & Turkel 2000). It used to be a consensus in the field that the generation problem of OT is NP-hard in the size of the grammar (e.g., Eisner 1997; 2000; Idsardi 2006a;b). This consensus was challenged by András Kornai in two squibs (2006a;b) that probably made one of the liveliest moments in the history of the *Optimality List* and the ROA *Rutgers Optimality Archive* (and see also Heinz et al. 2009).

Heuristic optimization algorithms, including simulated annealing and genetic algorithms, have been successfully deployed to find an approximately good solution for NP-hard problems (Reeves 1995, pp. 6–11). While they do not guarantee to always return *the* best solution, they do so reasonably well, returning the optimum pretty often, and otherwise returning a solution almost as good as the best one. Whether the generation problem in OT is NP-hard, or it is not, two further arguments can also be given for the use of heuristic optimization: similar trends in the cognitive sciences in general, beyond linguistics (e.g., Gigerenzer et al. 1999), and the very fact that our human speech production is also known to be prone to errors. Hence our interest in "less perfect" approaches and the motivation to employ *simulated annealing* for Optimality Theory (Bíró 2005a;b; Biró 2006).

Let me now summarize simulated annealing (in a way that is based on Biró 2007). Equations (1) and (10) define Optimality Theory as an optimisation problem. The task is to find the candidate x^* that optimizes $H(x)$.

Many heuristic algorithms do not always find the (globally) optimal candidate, but are simple and still efficient because they exploit the structure of the search space, which is the candidate set in our case. This structure is realized by a *neighborhood relation*: for each candidate x there exists a set `neighbors`(x), the set of the neighbors of x. It is often supposed that neighbors differ only *minimally*, whatever that means. The neighborhood relation is usually symmetric, irreflexive and results in a connected graph-like structure: any two candidates are connected by a finite chain of neighbors. More details of this relation should depend on the specific linguistic phenomenon under discussion.

The neighborhood structure – also called the *topology* – invites for a *random walk* in the search space, that is, on the candidate set. This walk can be conceived of as a series $x_0, x_1, x_2, \ldots, x_L$ of candidates. Candidate x_i, to be also referred to as the position of the random walker at time i, must be either identical to, or a neighbor of the candidate x_{i-1}, the previous position of the random walker. Position x_0 will be called the *initial position* (x_{init}), and x_L shall be the *final position* (x_{final}) of the random walk, whose length is L, the number of "steps".

```
ALGORITHM Gradient Ascent: OT with restricted GEN
 x := x_init;
 repeat
        x_prev := x;
        x      := most_harmonic_element( {x_prev} U neighbors(x_prev) );
 until x = x_prev
 return x                 # x is an approximation to the optimal solution
```

Figure 3: Gradient Ascent: iterated Optimality Theory with a restricted GEN (Do-α)

```
ALGORITHM Randomized Gradient Ascent
 x := x_init ;
 repeat
         Randomly select x' from the set neighbors(x);
         if  (x' not less harmonic than x)    then   x := x';
 until stopping condition = true
 return x                 # x is an approximation to the optimal solution
```

Figure 4: Randomized Gradient Ascent

A random walker, such as a hedgehog, will walk in a *landscape*. The landscape's horizontal map is provided by the neighborhood structure, whereas its vertical dimension is the objective function H to be optimized. The hedgehog's goal is to climb the highest point in this landscape.

The simplest algorithm, *gradient ascent*, comes in two flavors. The version in Fig. 3 defines x_{i+1} as the best element of the set $\{x_i\} \cup \texttt{neighbors}(x_i)$. The hedgehog walks as long as x_{i+1} differs from x_i, and the algorithm is deterministic for each x_{init}. This kind of optimization has been known in Optimality Theory since 1993 (Prince & Smolensky 1993; 2004) as *serial evaluation* (McCarthy 2007) or *harmonic serialism* (McCarthy 2010): x_{init} is the underlying form, Do-α (a restricted version of Gen) creates the set $\{x\} \cup \texttt{neighbors}(x)$, whereas the Eval module finds its best element in each iteration.

The second version of *gradient ascent* is stochastic (Fig. 4). In step i, the hedgehog chooses a random $x' \in \texttt{neighbors}(x_i)$, using some predefined probability distribution on this set (often a uniform distribution). If neighbor x' is not worse than x_i, then the next element x_{i+1} of the random walk will be x'; otherwise, x_{i+1} is x_i. The stopping condition requires the number of iterations to reach some sufficiently large value, or the average improvement of the objective function in the last few steps to drop below a threshold (usually zero). Then the algorithm returns the output x_{final}, which is likely to be a local optimum.
Simulated annealing (Fig. 5) plays with this second theme to increase the hedgehog's chances of finding the global optimum and avoid being trapped in unwanted local optima. The idea is the same, but if x' is worse than x_i, then there is still a chance to move to x'. Importantly, however, this probability is reduced to 0, as the algorithm proceeds. (In some versions of simulated annealing, which we ignore here, if x' is better than x_i, the chance of moving to x' is less than 1, with this probability gradually converging to 1.)

The *transition probability* of moving to x' depends on the objective function H at points x_i and x', as well as on a parameter of the algorithm, $T > 0$, called *temperature* for historical reasons (Metropolis et al. 1953; Kirkpatrick et al. 1983; Černý 1985):

$$P(x_i \to x'|T) = e^{\frac{H(x') - H(x_i)}{T}}. \tag{11}$$

(Note that usually an energy function E is minimized, and not a harmony function H maximized, and therefore the standard formula also includes a negative sign in the exponent.) If the randomly chosen neighbor x' is less harmonic than x (if $H(x') < H(x)$), then a random number r is generated, and we move to x' if and only if $r < P(x_i \to x'|T)$.

```
ALGORITHM: Simulated Annealing
 Parameters:  x_init # initial state (often randomly chosen)
              T_max  # initial temperature > 0
              alpha  # temperature reduction function = cooling schedule

 x := x_init ;
 T := T_max  ;
 Repeat
     Randomly select x' from the set neighbors(x);
     Delta :=  H(x') - H(x) ;
     if ( Delta > 0 ) # neighbor is more harmonic than current position
      then
       x := x' ;
      else
       # move to x' with transition probability P(Delta;T)=exp(Delta/T):
       generate random r uniformly in range (0,1) ;
       if ( r < exp ( Delta / T ) )
                  then    x := x' ;
      end-fi
     end-fi
     T := alpha(T)   # decrease T according to cooling schedule
 Until stopping condition = true
 Return w            # w is an approximation to the optimal solution
```

Figure 5: *Maximizing* a real-valued harmony function $H(x)$ with simulated annealing

Temperature T is gradually decreased following a *cooling schedule*, a decreasing series of values for T, so that in step i, the value of the temperature is T_i:

$$T_{max} = T_0 > T_1 > T_2 > \ldots > T_i > \ldots > T_L = T_{min} > \text{ but close to 0.} \qquad (12)$$

Some allow the same T_i value to be re-employed a finite number of times *rep*, independent of (Reeves 1995, 26), or dependent on (Henderson et al. 2003) T_i.

As the temperature T decreases, the exponent in (11) becomes an increasingly low negative number, and the transition probability $P(x_i \rightarrow x'|T)$ converges to zero. With very low temperatures, the hedgehog would not move towards lower harmony anymore.

The final position of the random walker will be returned as the output of simulated annealing. It is a stochastic algorithm without the guarantee of always finding the global optimum. In fact, the hedgehog may be stuck in a local optimum, having climbed a hill that is not the highest in the landscape, but the temperature being too low already for the hedgehog to take a counter-optimal step. The probability of returning the global optimum will be referred to as the *precision* of the algorithm.

An important fact about simulated annealing is that precision can be made to converge to 1 as the number of iterations L grows (Reeves 1995; Henderson et al. 2003). The next section illustrates why this happens so. Smolensky & Legendre (2006) repeatedly refer to this fact as an advantage of their proposal: if the language model is given ample time, it will almost certainly find the most harmonic, that is, the grammatical form. (Note, though, that both the formal analysis and the practice of simulated annealing know of cases that require very long running times; cf. Reeves 1995, 63.) Biró (2006) takes a different approach: the language model unable to find the grammatical form makes a performance error, similarly to the human brain. Moreover, if the algorithm is given less time, it makes more errors, like a fast speaking human. Speed can be traded for precision, and fast speech can be modeled with simulated annealing (Bíró 2005b).

So far, we have only discussed simulated annealing with a real-valued harmony function. However, I have earlier proposed a way to adopt simulated annealing (Fig. 5) to OT; in particular, I had to adapt the exponential expression in (11) to the non-real valued objective function in (9) (Bíró 2005a;b; Biró 2006). While I have presented various mathematical arguments that lead to the same *Simulated Annealing for Optimality Theory* (SA-OT) *Algorithm* (Biró 2006, chapters 2 and 3; and a different train of thought in Biró 2013 that could also be applied to SA-OT), critics may still argue that the SA-OT Algorithm is removed from “real” simulated annealing.

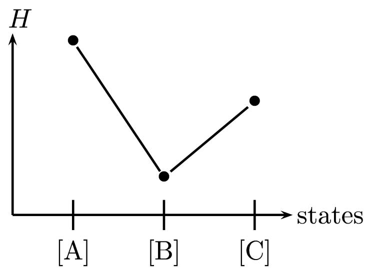


Figure 6: V landscape, with three candidates in a row. [B] in the center is worse than the two candidates on the peripheries. [A] is the global optimum, while [C] is locally optimal.

Table 3: A possible tableau for the asymetrical V shaped landscape. Note symbol ∼ marking the local optimum.

	C_2	C_1
☞ [A]		
[B]	*	
∼ [C]		*

In particular, SA-OT lacks the above mentioned property of simulated annealing: in some cases increasing the number of iterations did not improve the precision of the algorithm (refer to Biró 2006, sections 2.3.2 and 6.4, as well as Biró 2009). It has been argued that the reason is *strict domination*. Look at the landscape in Fig. 6 with tableau 3. This toy grammar includes two local optima separated by the suboptimal candidate [B]. [A] is the global optimum, better than [C] due to low ranked constraint C_1. However, in simulated annealing, non-adjacent candidates are not compared directly, and their frequencies in production emerge as a consequence of the landscape. (This is a significant difference between SA-OT and Maximum Entropy OT, cf. Goldwater & Johnson 2003.) Both [A] and [C] defeat [B] by the highly ranked constraint C_2, and strict domination requires that lower ranked constraints do not play any role. Consequently, as shown in Biró (2006, section 2.3.2), SA-OT will return both [A] and [C] with a probability of 50%, independently of the speed of the algorithm.

Without entering further details of the SA-OT Algorithm, the main thrust of the current paper is to explain that this non-convergence property is intrinsic to Optimality Theory: it appears not only in the SA-OT Algorithm, which adopts strict domination *a priori*, but also in the strict domination limit of applying simulated annealing to symbolic HG. In other words, strict domination grammars, which practically do not count, will make errors, even if ample time is available for them to perform computations.

5. Why does simulated annealing work?

5.1. A simple model and its dynamics

First, let us try to understand why standard simulated annealing is successful as an optimization algorithm: why its precision converges to 100% if the length of the random walk is increased. The key will be that in a certain phase of the algorithm (the "second phase") hedgehogs can escape local optima and be attracted to the global optimum.

Let us return to the simplest search space with a global optimum and another local optimum (Fig. 6), which will be referred to henceforth as an 'asymmetric V landscape'. Our randomly walking hedgehog in state [B] has two neighbors to choose from. Suppose that both neighbors have a chance of 0.5 to be picked. Then, either [A] or [C] is chosen, and the hedgehog will move there with a transition probability $P(B \rightarrow A, C|T) = 1$, because $H(B)$ is lower than both $H(A)$ and $H(C)$. If the random walker is, however, in state [A] or [C], the only neighbor has a reduced probability of being moved to. Following eq. (11), the chances are:

$$\begin{aligned} p_A(T) := P(A \rightarrow B|T) &= e^{\frac{H(B)-H(A)}{T}} \\ p_C(T) := P(C \rightarrow B|T) &= e^{\frac{H(B)-H(C)}{T}} \end{aligned} \tag{13}$$

Since $H(A) > H(C)$, we can easily see that $p_A(T) < p_C(T)$ at any time, at any temperature T. As a side remark, this inequality does not hold in SA-OT in the case presented by tableau 3, and that is why the global optimum [A] is not able to attract the random walker away from the other local optimum. Hence, the 50% precision of SA-OT.

Now suppose that many hedgehogs walk simultaneously. The average number of hedgehogs moving from state x to a neighboring state x' in some iteration of the algorithm is the product of the number of hedgehogs in state x, the probability of choosing neighbor x' when at x, and the transition probability $P(x \rightarrow x'|T)$. At time step i, let a_i, b_i and c_i denote the number of random walkers in states [A], [B] and [C] respectively. By using the above probabilities to calculate the flows from and into each state, we obtain:

$$\begin{aligned} a_{i+1} &= a_i - a_i \cdot p_A(T_i) + \frac{1}{2} b_i \\ b_{i+1} &= b_i - \frac{1}{2} b_i + a_i \cdot p_A(T_i) - \frac{1}{2} b_i + c_i \cdot p_C(T_i) \\ c_{i+1} &= c_i - c_i \cdot p_C(T_i) + \frac{1}{2} b_i \end{aligned} \tag{14}$$

As expected, the number of hedgehogs $a_i + b_i + c_i = N$ is constant in time. The precision of the algorithm is the probability that a random walker finishes the walk in the globally optimal state [A], which is a_∞/N. In what follows we discuss the mechanisms that increase this precision.

5.2. Extreme temperatures, and in between

Let us now consider two extreme cases. First, we suppose that temperature is very low from time step $i = 0$ onwards, that is, $T \ll H(C) - H(B)$. Then, according to (13), $p_A \approx 0$ and $p_C \approx 0$. In this case, the poor hedgehogs are unable to escape from the local optima. The initial population b_0 in state [B] is distributed within one step between the two local optima, and from this point onwards $a_i = a_0 + 0.5b_0$ and $c_i = c_0 + 0.5b_0$ for all $i > 0$. In brief, the random walkers are frozen into the local optima at very low temperatures.

In the second extreme case, temperature is very high: $T \gg H(A) - H(B)$, resulting in $p_A \approx 1$ and $p_C \approx 1$, from time step $i = 0$ onwards. Then the random walkers oscillate between the central and the peripheral positions. Since the random walkers are equally distributed by position [B], the whole system itself oscillates between two states. At odd time steps $a_{2i-1} = c_{2i-1} = 0.5b_0$ and $b_{2i-1} = a_0 + c_0$, whereas at even time steps $a_{2i} = c_{2i} = 0.5b_1 = 0.5(a_0 + c_0)$ and $b_{2i} = b_0$ $(i \geq 1)$. Even if initially $a_0 \neq c_0$, a short period with extremely high temperatures will result in $a_i = c_i$.

In the practice of simulated annealing, temperature T drops from a very high to a very low value in many steps. In the *first phase*, $T \gg H(A) - H(B)$, while in the *third phase* $T \ll H(C) - H(B)$. In what I shall call the *second phase*, temperature T is, informally speaking, "in the magnitudes of" $H(A) - H(B)$ and $H(C) - H(B)$.

We have just seen that by the end of the first phase $a_I = c_I$. The precision of the algorithm will be determined by $a_{II} - c_{II}$ at the end of the second phase, since in the third phase the states get frozen. If a_{II}, b_{II} and c_{II} denote the population in each of the states after the second and before the third phase, then the precision of the algorithm is $\frac{a_{II}+0.5b_{II}}{N}$. Obviously, here the phases are idealized, and in reality their boundaries are not so clear. Yet, this idealization will contribute to our better understanding of simulated annealing.

Next, observe that from equations (14):

$$a_{i+1} - c_{i+1} = (a_i - c_i) + (c_i \cdot p_C(T_i) - a_i \cdot p_A(T_i)) \qquad (15)$$

At the beginning of the second phase $a_I = c_I$, and therefore:

$$a_{I+1} - c_{I+1} = (a_I - c_I) + a_I(p_C(T_I) - p_A(T_I)) \qquad (16)$$

Population a_i and c_i will begin to diverge if and only if there is a period during the simulation when $p_A(T_i) \neq p_C(T_i)$. Otherwise, $a_i - c_i$ remains constantly zero.

For instance, if $H(A) = H(C)$, then $p_A(T_i) = p_C(T_i)$ at all times. Therefore, independently of the original distributions, simulated annealing will return both states [A] and [C] in 50% of the cases.

Even if $H(A) > H(C)$, SA-OT offers no such period in Fig. 6 and tableau 3: due to reasons related to strict domination, the transition probabilities only depend on the fatal constraint (called the "highest uncanceled violation mark" by Prince & Smolensky 1993; 2004), which does not distinguish between $p_A(T_i)$ and $p_C(T_i)$. On the contrary, when standard simulated annealing is applied to symbolic harmonic grammar, eqs. (13) ensure that $p_C(T) > p_A(T)$. So $a_i - c_i$ can turn positive at the beginning of the second phase. The higher $p_C(T_i) - p_A(T_i)$, the quicker the divergence between a_i and c_i. Moreover, the longer this period, the higher the precision of the algorithm. These are the two mechanisms that contribute to the success of simulated annealing, to its high precision.

By way of example, let us suppose that there is a period when $H(A) - H(B) \gg T \gg H(C) - H(B)$, that is when $p_A(T) \approx 0$ and $p_C(T) \approx 1$. Then, state [A] acts as a trap for the hedgehogs, while it is still possible to escape from [C]. A hedgehog will end up in state [C] only if whenever he is in [B], he decides to move to [C], and not to [A], which has a probability of 0.5 each time. Provided that this period of the algorithm lasts $2k$ iterations, our hedgehog ends up in [C] with a probability of 0.5^k, and in [A] with a probability of $1 - 0.5^k$. Consequently, the more iterations in this crucial phase of the simulation, the higher the precision of the algorithm. The precision converges to 100% as $2k$ grows.

This is the main idea behind simulated annealing, even if details are more complicated, even for this simple landscape. For instance, after a_i and c_i have diverged, $c_i \cdot p_C(T_i) - a_i \cdot p_A(T_i)$ does not need to stay positive in (15), and so $a_i - c_i$ will not necessarily grow forever. If simulated annealing is very ("infinitely") slow, the system may reach an equilibrium in which $a_i \cdot p_A(T_i) = c_i \cdot p_C(T_i)$. That will be the topic of the next subsection.

To summarize, in the *first* and *second* phases, the hedgehogs can escape local optima. In the *second* and *third* phases, the hedgehogs are attracted by the global optimum. The longer the *second* phase (that is, the more iterations are available in the second phase), the greater the chance that a random walker will end up in the global optimum.

5.3. Equilibrium of our system

Candidates [A], [B] and [C] have been called states, in which each of the N hedgehogs can be found. As opposed to these *states*, the whole system can be characterized with some *macrostate*: following the physical analogy, a macrostate of the whole system is a distribution (a_i, b_i, c_i) of the random walkers. Different random walkers can change their state, hence the *microstate* of the whole system can change (a third concept); yet, the macrostate does not alter as long as the overall distribution remains the same.

A macrostate is an *equilibrium state* if it does not change in time, that is, $a_{i+1} = a_i$, $b_{i+1} = b_i$ and $c_{i+1} = c_i$. Maybe no random walker moves: the system is frozen, thus the microstate is also invariable. But it can very much be the case that individual hedgehogs move from one state to another one, but the distribution remains the same. From eqs. (14) and (15) we conclude that the system is in equilibrium if and only if

$$\begin{aligned} a_i \cdot p_A(T) &= c_i \cdot p_C(T) \\ b_i &= a_i \cdot p_A(T) + c_i \cdot p_C(T) \end{aligned} \tag{17}$$

If enough iterations are performed, then the system can converge to this state. Decreasing the temperature very ("infinitely") slowly allows the system to stay in this macrostate of equilibrium. Then

$$\frac{c_i}{a_i} = \frac{p_A}{p_C} = e^{\frac{H(B)-H(A)}{T} - \frac{H(B)-H(C)}{T}} = e^{\frac{H(C)-H(A)}{T}} \tag{18}$$

will steadily hold true. Gradually decreasing T to zero, results in $\frac{c_i}{a_i}$ also converging to zero. The larger the difference $H(A) - H(C)$, the faster this convergence. In sum, an "infinitely slow" annealing (parameter T decreased to zero in many-many steps) is characterized by a precision of 1: it will only return the global optimum [A], and never the other local optimum, [C].

As a side note, solving the equation system (14) in the fixed point – that is, for $a_{i+1} = a_i = a^*$, $b_{i+1} = b_i = b^*$ and $c_{i+1} = c_i = c^*$ – with $N = a^* + b^* + c^*$ leads us to

$$\begin{aligned} a^*(T) &= N \frac{p_c}{p_c + 2p_c p_c + p + a} \\ b^*(T) &= 2N \frac{p_a p_c}{p_c + 2p_c p_c + p + a} \\ c^*(T) &= N \frac{p_a}{p_c + 2p_c p_c + p + a} \end{aligned} \tag{19}$$

In turn, inserting the transition probabilities (13) into (19) yields a Boltzmann distribution (in fact, a Boltzmann distribution in which [B] corresponds to a degenerate state, that is, to two states that have the same harmony and have been collapsed into one) as the state of equilibrium:

$$\begin{aligned} a^*(T) &= \frac{N}{Z(T)} e^{\frac{H(A)}{T}} \\ b^*(T) &= 2\frac{N}{Z(T)} e^{\frac{H(B)}{T}} \\ c^*(T) &= \frac{N}{Z(T)} e^{\frac{H(C)}{T}} \end{aligned} \tag{20}$$

where N is the number of random walkers run in parallel, whereas

$$Z(T) = e^{\frac{H(A)}{T}} + 2 \cdot e^{\frac{H(B)}{T}} + e^{\frac{H(C)}{T}} \tag{21}$$

is called the *partition function.* As $T \to +0$, the largest term (the first one) will dominate the partition function, and therefore $\lim a^*(T) = N$, but $\lim b^*(T) = 0$ and $\lim c^*(T) = 0$.

Observe that *Maximum Entropy OT* (Goldwater & Johnson 2003) postulates a distribution similar to (20) (ignoring the factor 2 in b^* and Z), as if annealing stopped at a positive value of temperature T.

The next section derives the main result of this paper. It shows that strict domination – postulated by Optimality Theory, and an asymptotic case in q-HG – allows cases in which $p_C(T) - p_A(T)$ is 1 in a crucial phase of the simulation (hence, simulated annealing is maximally efficient); but also cases in which $p_C(T) - p_A(T)$ is constantly 0, and therefore simulated annealing produces *irregular forms.*

6. Simulated annealing in the strict domination limit

6.1. Simulated annealing for *q*-HG

The present section contains the core message of this paper by asking what the consequences are of the $q \to +\infty$ *strict domination limit* for simulated annealing when it is applied to a q-HG grammar. We shall observe that strict domination can lead to very efficient computation, but also to severe errors.

Increasing q will increase the range of the objective function $H(w)$. It will also magnify the differences $H(x') - H(x)$ in the equation of the transition probability (11).

It follows that the *cooling schedule* must also be adapted. If the cooling schedule remained the same, even the highest temperatures would become very low compared to the differences in the objective function at high q values. In the strict domination limit, the algorithm would therefore miss its crucial first two phases, and start immediately with randomized gradient descent (Fig. 4), instead of simulated annealing (Fig. 5).

Therefore, the cooling schedule should be made a decreasing series of functions of q:

$$T_{max}[q] = T_0[q],\ T_1[q],\ T_2[q],\ \ldots,\ T_i[q],\ \ldots,\ T_L[q] = T_{min}[q] > 0 \tag{22}$$

Additionally, a cooling schedule will satisfy two requirements: first, we require that for some reasonable q_0 and for any $q > q_0$: $T_i[q] > T_{i+1}[q]$. Second, we also posit $\lim_{q\to+\infty} T_i[q] = +\infty$ for any i.

More specifically, the cooling schedule should be such that the three phases discussed in the previous section should be discernible. For any q, the first values in the series should be "much greater" than any possible difference in harmony of two neighboring candidates; which, based on (8), is in the order of magnitude of q^n. Similarly, the last values in the series should be for any q "much smaller" than the smallest possible difference in harmony, which is q, one violation difference of the lowest ranked constraint.

Biró (2006) suggested using $T_i[q] = t_i \cdot q^{K_i}$, where K_i was decreased in an outer loop (from K_{max} to K_{min}, using K_{step}), and for each K_i, t_i was decreased from t_{max} to t_{min} by t_{step}. For instance, the case $K_{max} = 4$, $K_{min} = 0$, $K_{step} = 1$, $t_{max} = 3$, $t_{min} = 0.5$ and $t_{step} = 0.5$ would look like:

$$3 \cdot q^4,\ 2.5 \cdot q^4,\ 2 \cdot q^4, \ldots, 0.5 \cdot q^4,\ 3 \cdot q^3, \ldots, 0.5 \cdot q^3,\ 3 \cdot q^2, \ldots, 0.5 \cdot q^0 \tag{23}$$

This kind of cooling schedule will be referred to as *linear*. Another option is to diminish the temperature exponentially (Biró 2009):

$$T_i = (c \cdot q^n)^{\frac{m-i}{m}} \tag{24}$$

where n is the number of constraints in the q-HG grammar (the exponent of the highest ranked constraint). As one violation of the highest ranked constraint contributes q^n to the harmony function in (8), a large c (e.g., $c = 100$ – supposing that neighbors differ in only a few violations of constraint C_n) guarantees that initially the random walker will move freely: for any x and x', the transition probability $P(x \to x'|T_0) \approx 1$. At the same time, parameter m determines the speed of the cooling schedule: a large m diminishes the temperature only slowly. By the mth step, temperature is reduced to $T_m = 1$. The smallest possible difference in harmony – corresponding to a single violation of the lowest ranked constraint C_1 – is q. Therefore, if q is large, then $|H(x) - H(x')| \gg T_m$, which means that after m steps the system will have been frozen, the algorithm will have reached its third phase.

6.2. The V landscape: the good case

In order to understand the behavior of simulated annealing applied to q-Harmonic Grammar in the strict domination limit, let us return to the V landscape (Fig. 6). Recall equation (15), repeated here:

$$\begin{aligned} a_{i+1} - c_{i+1} &= (a_i - c_i) + (c_i \cdot p_C(T) - a_i \cdot p_A(T)) \\ &= (a_i - c_i)(1 - p_A(T)) + c_i(p_C(T) - p_A(T)) \end{aligned} \quad (25)$$

Remember that at the beginning of the second phase $a_i = c_i$. The speed at which the number of random walkers in states [A] and in [C] will diverge during the second phase therefore depends on $p_C(T) - p_A(T)$. If this value is close to zero, then the divergence will be very slow, and only an extremely large number of iterations can guarantee finding the globally optimal state with a high probability. If, however, there is a phase in the simulation (there is a value T) when $p_A(T)$ and $p_C(T)$ are very different, then the algorithm will be efficient.

The conclusion thus has been that the efficiency of simulated annealing depends crucially on the phase in which $p_A(T)$ is low and $p_C(T)$ is high. Is there such a phase in the strict domination limit? We shall see that in certain cases the strict domination limit makes simulated annealing extremely efficient, but not in other cases.

Let us start with the good case. Consider the following tableau for the V landscape (Fig. 6; $\alpha > \beta$, and both are positive integers):

	C_α	C_β
[A]		*
[B]	*	*
[C]	*	

(26)

Suppose that all other constraints do not distinguish between the three candidates, and so they contribute the same constant term τ to the harmony. The harmony of the states are as follows: $H(A) = \tau - q^\beta$, $H(B) = \tau - q^\alpha - q^\beta$, and $H(C) = \tau - q^\alpha$. Consequently:

$$
\begin{aligned}
p_A(T) = P(A \to B|T) &= e^{-\frac{q^\alpha}{T}} \\
p_C(T) = P(C \to B|T) &= e^{-\frac{q^\beta}{T}}
\end{aligned} \tag{27}
$$

What we need is a cooling schedule with a second phase in which $p_C(T) - p_A(T)$ is large. A useful cooling schedule will start with $T_{max}[q] \gg q^\alpha$ and end with $T_{min}[q] \ll q^\beta$. By having a sufficient number of intervening steps, there will be some $T_i[q] = q^\gamma$ where $\alpha > \gamma > \beta$. Such a cooling schedule can easily be constructed. In the case of a linear cooling schedule (23), use $t_{\max} = t_{\min} = 1$, $K_{\max} = \alpha + 0.5$ and $K_{\text{step}} = 1$. Alternatively, use $K_{\text{step}} < \alpha - \beta$, to make sure some $T_i[q] = t_i \cdot q^{K_i}$ falls between q^α and q^β. If you prefer the exponential cooling schedule scheme (24), then $m > n$ will make the exponent of q take some value between any two adjacent integers.

In turn, employing any of these cooling schedule schemes, let i be such that $T_i[q] = t_i \cdot q^\gamma$ with $\alpha > \gamma > \beta$. In this case,

$$
\begin{aligned}
\lim_{q\to+\infty} p_A(T_i[q]) &= \lim_{q\to+\infty} e^{-\frac{q^\alpha}{t_i \cdot q^\gamma}} = 0 \\
\lim_{q\to+\infty} p_C(T_i[q]) &= \lim_{q\to+\infty} e^{-\frac{q^\beta}{t_i \cdot q^\gamma}} = 1
\end{aligned} \tag{28}
$$

Consequently, $p_C(T_i[q]) - p_A(T_i[q])$ converges to 1 in the strict domination limit. For large q, at iteration i, our random walking hedgehog is free to leave the locally optimal state [C], but is stuck in the global optimum [A].

This situation was already discussed in section 5.2, and we saw that the probability of ending up in [A] could be made to converge to 1 by increasing the number of steps in this phase of the algorithm. This can be achieved, for instance, by reducing the value of the t_{step} parameter in a linear cooling schedule (23), or by increasing m in an exponential cooling schedule (24). If $t_{step} < \frac{t_{max}-t_{min}}{2k}$, or if $m > \frac{2kn}{\alpha-\beta}$, then the algorithm will spend at least $2k$ iterations such that $q^\alpha > \mathcal{O}(T_i[q]) > q^\beta$, corresponding to a precision of at least $1 - 0.5^k$ in the strict domination limit.

Strict domination in this case has proven to be an asset. Increasing q also increases $p_C(T_i[q]) - p_A(T_i[q])$, and so simulated annealing is expected to work better.

6.3. The V landscape: the bad case

The situation will be very different with the following tableau (again, $\alpha > \beta$, and both are positive integers):

	C_α	C_β
[A]		
[B]	*	*
[C]		*

(29)

This time, $H(A) = \tau + 0$, $H(B) = \tau - q^\alpha - q^\beta$ and $H(C) = \tau - q^\beta$, whence

$$\begin{aligned} p_A(T) = P(A \to B|T) &= e^{-\frac{q^\alpha + q^\beta}{T}} \\ p_C(T) = P(C \to B|T) &= e^{-\frac{q^\alpha}{T}} \end{aligned} \tag{30}$$

What is $p_C(T) - p_A(T)$ in the strict domination limit?

$$\begin{aligned} \lim_{q\to+\infty}(p_C(T) - p_A(T)) &= \lim_{q\to+\infty} e^{-\frac{q^\alpha}{T}}\left(1 - e^{-\frac{q^\beta}{T}}\right) = \\ &= \lim_{q\to+\infty} e^{-\frac{q^\alpha}{T}} \cdot \lim_{q\to+\infty}\left(1 - e^{-\frac{q^\beta}{T}}\right) \end{aligned} \tag{31}$$

This limit is always zero. Namely, if $T[q] < \mathcal{O}(q^\alpha)$, that is, if $\lim \frac{q^\alpha}{T[q]} = \infty$, then the first limit is zero and the second limit is less than or equal to 1. If, on the other hand, $T[q] > \mathcal{O}(q^\beta)$ (that is, $\lim \frac{q^\beta}{T[q]} = 0$), than the second limit is zero and the first limit is less than or equal to 1.

Thus, when simulated annealing is applied to a V landscape with tableau (29), the difference $p_C - p_A$ stays zero in the strict domination limit, at *any* temperature. The consequence of this fact for the dynamics in eq. (25) is that the probability of a hedgehog to be in state [A] or in state [C] will never diverge, yielding a 50% precision for *all* cooling schedules.

In summary, we have analyzed two variants of the asymmetric V landscape, displaying different behaviors in the strict domination limit. As the parameter q of a q-HG grammar is gradually increased, so does the behavior of simulated annealing approach the behavior of SA-OT. In the case of tableau (26), the precision converges to 100% as the number of iterations grows in the second phase; but it stays 50%, independently of the cooling schedule, for tableau (29). Next, we confirm this analysis with computer experiments.

7. Experiments with the V landscape

It is always good practice to also support the conclusions of an analytical discussion with computer experiments. Therefore, this section reports the results of simulations run in a V landscape with three states (candidates), as shown in Fig. 6. Can we confirm the above analyses of the grammars in tableaux (26) and (29)?

For the sake of concreteness, the two constraints were assigned weights q^2 and q respectively (i.e., $\alpha = 2$ and $\beta = 1$). Note that in both tableaux, the relative harmony of the three candidates are independent of q (*viz.*, $H_q([\text{A}]) > H_q([\text{C}]) > H_q([\text{B}])$ for all q), not displaying any kind of cumulativity. Thus, the grammatical output is always [A].

The Java implementation of the simulated annealing algorithm in Fig. 5 was run on the *Atlasz HPC cluster* of the ELTE university. For each parameter combination discussed below, 10^6 random walks were launched, so that we could measure the *precision* of the algorithm by counting the frequency of returning the globally optimal candidate. With such a large sample size, the standard error of the population proportion (i.e., the precision) is below 10^{-3}. We also measured the distribution – mean and standard deviation – of the *length of the random* walk, that is, the number of iterations until convergence.

The three candidates were used by turns as the initial position of the random walk. The exponential cooling schedule followed (24), with variable i (initially 0) increased by 1 in each iteration. Parameters c and n were fixed: $c = 100$, to ensure a very high initial temperature, and $n = 2$, corresponding to the two constraints in the grammar. A step by the random walker increasing the violation of the higher ranked constraint decreases the harmony by q^2; when the temperature is initially $T_0 = c \cdot q^2$, even such a step has a probability of $\exp(-1/c) \approx 1$. After m steps, temperature dropped to $T_m = 1$; this point in time can be roughly seen as the beginning of the third phase, when temperature has become much lower than the smallest possible difference in harmony, q. Increasing parameter c increases how many of the first m iterations "kind-of" belong to the first phase, whereas decreasing parameter c increases the number of iterations in the "second phase". Remember that the success of the algorithm depends on the number of iterations in this second phase. In accordance with our prediction, simulations confirm that choosing a smaller c slightly improves the precision of the algorithm.

The stopping condition required the random walker not to move for $\ell = 60$ iterations. Recall that if the random walker is in position [B], it will always move to one of its neighbors. It will not move if it is in a local optimum, and the random number generated is higher than the transition probability (11). For this to happen 60 times, we must be extremely unlucky, unless the transition probability is already extremely low, as the consequence of a very low temperature. Reducing parameter ℓ to 40 or 20 will marginally decrease the precision of the algorithm and the average length of the random walk. Reducing it further to 10 will result in a more significant loss in precision, accompanied by an average length of the random walk diminished by a few steps. Indeed, given the very large sample size, it is not unexpected that a few times the random number generator will produce ten consecutive large values, stopping the algorithm prematurely.

For each of the two grammars analyzed above, the "good case" and the "bad case", we report in details the effect of tuning the two most interesting parameters: the base q, and m, the speed of simulated annealing. Tables 4 and 6 in the Appendix present the precision for each parameter combination. Given the sample size of 10^6 for this binary process (the output being either "correct" or "incorrect"), the standard error of the sample proportion, but also the error bar for the estimated proportion is below 0.1%.

Tables 5 and 7 present the speed of convergence: for each parameter combination, the mean and standard deviation of the 10^6 simulation lengths. Simulation length refers to the number of iterations until the random walker got stuck in the global or another local optimum: the value of the variable i in the cooling schedule (24) when the stopping condition becomes true minus ℓ, the number of iterations the random walker has been stuck here. Again due to the large sample size, the standard error of the mean in each cell is smaller by three orders of magnitude than the reported standard deviation.

The number of iterations is comparable to m, while it significantly decreases as q increases. An empirical law of the form $\textit{nr of iterations} = m^{\chi} \cdot (\ln q)^{\phi}$ approximates the observed data reasonably well, even though a closer look at the tables reveals a more complex behavior. Fitting the output of a separate set of experiments, we obtained $\chi = 0.954$ and $\phi = -0.351$ for the "good case", and $\chi = 0.945$ and $\phi = -0.361$ for the "bad case" (the difference between the two cases is highly significant). The conclusion is clear: simulated annealing with a given cooling schedule – in our case, a specific value of the parameters c and m – becomes faster in the strict domination limit.

Returning to precision, the plots in Fig. 7 present the results of yet another set of simulations, each data point measured with 10^6 runs. Similarly

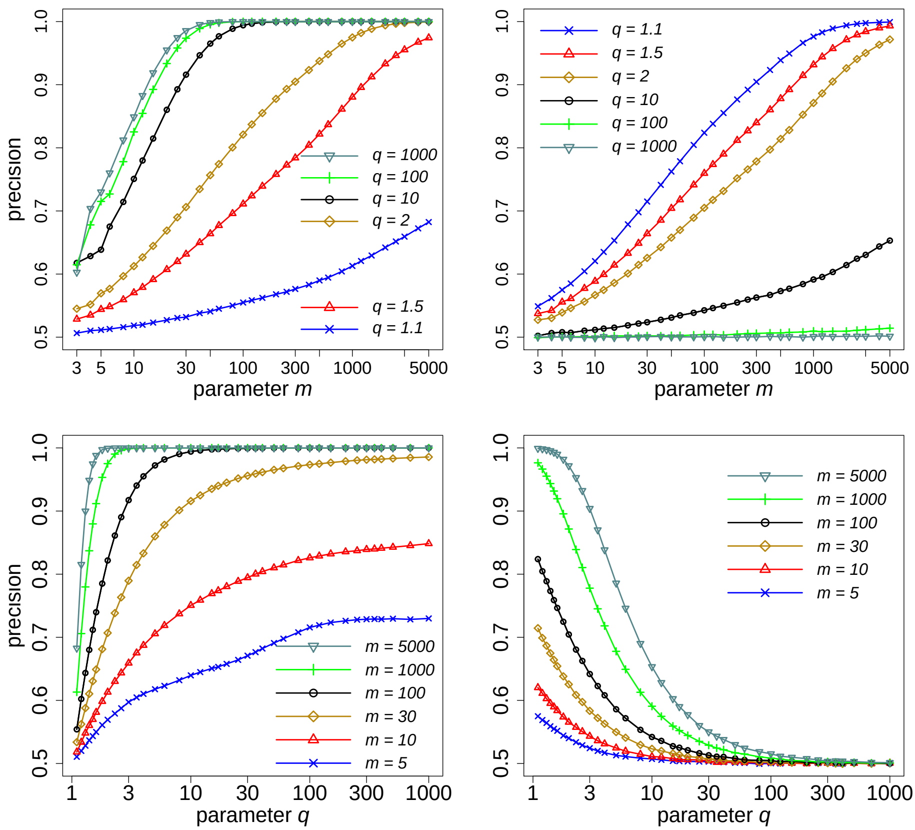


Figure 7: Precision of simulated annealing q-HG, as a function of parameters m and q, for the two grammars discussed: (26) shown on the left panels, and (29) on the right panels. For a given q, increasing m always improves precision. However, as q grows from 1.1 to 1000 (on a logarithmic scale), the two grammars display opposite behaviors.

to Tables 4 and 6, we can observe how for a given q, increasing the number of iterations (increasing m) improves precision. This behavior is not at all surprising, as a q-HG grammar is a real-valued optimization problem. From the general convergence properties of "standard" simulated annealing, we know that a sufficiently slow cooling schedule will produce high precision. Mathematically speaking, I conjecture that for any q and all $\epsilon > 0$ there exists an m such that a specific q-HG grammar will yield a precision higher than $1 - \epsilon$.

Yet, this statement with a reversed scope is not necessarily true. Observe the plots as q grows. In the case of the "good grammar" on the left panels, the strict domination limit corresponds to some precision between 50% and 100%, depending on m. It reminds us of the precision of SA-OT with the same tableau, which also depends on the cooling schedule. A large q paired with a large m easily yields a precision sufficiently close to 1. But such is not the case on the right panels, corresponding to the "bad grammar".

The formal analysis in the previous section and the current experimental results both suggest that no cooling schedule is good enough for all q-HG grammars based on the "bad case" tableau (29). In fact it seems that for any cooling schedule – probably even beyond the exponential cooling schedule scheme (24) – a sufficiently large q will yield a precision close enough to the exactly 50% precision observed for the SA-OT Algorithm:

Conjecture 3. *For any cooling schedule and for all* $\epsilon > 0$*, there exists a* $q_0 > 1$ *such that for all* $q > q_0$ *the precision of a q-HG grammar with (29) is less than* $0.5 + \epsilon$*.*

8. Summary: why is it human to err?

From the old joke with the three greengrocers we learn that optimizing locally is more convenient for the human brain than optimizing globally. But then, the hedgehog in the optimization procedure may climb the wrong hill, producing an error. Therefore, we conclude that to err is human.

We have compared simulated annealing with a real-valued harmony function, as it happens in connectionist and symbolic harmonic grammars, to simulated annealing with strict domination. In the former case, one can choose a sufficiently slow cooling schedule so that the precision of the algorithm (the probability of returning the global optimum) be greater than $1 - \epsilon$: the precision can be made to converge to 100%. This is not the case with strict domination, however. With some grammars – i.e., constraint hierarchies, and candidate sets with neighborhood structures – the precision of the *Simulated Annealing for Optimality Theory Algorithm* (SA-OT) does not converge to 1. The same applies to q-HG: if q is large enough, the precision can be far away from 100%.

Encouraged by Newmeyer (1983) and slightly diverging from standard terminology, I suggest using the phrase *grammatical form* for linguistic forms predicted by a grammar, such as the global optimum in OT-style frameworks (1). A grammar is a model of the native speaker's linguistic competence, "the speaker-hearer's knowledge of [their] language" (Chomsky 1965); whereas the implementation of this grammar should mirror the speaker's linguistic performance (Smolensky & Legendre 2006; Biró 2006).

Jackendoff (2007, 27) explains Chomsky's "knowledge" as "whatever is in speaker's heads that enables them to speak and understand their native language(s)". But what is in one's head? A network of neurons. Hence, the motivation to bridge connectionist harmonic grammars to symbolic ones, and then to OT grammars, via q-HG. Now, should a grammar be an adequate description of the speaker's knowledge, the grammar will correctly predict the forms produced and judged as acceptable – provided a perfect implementation thereof.

In an imperfect implementation, however, errors occur: forms that are not grammatical, but are nevertheless produced. These could be called *performance errors*. Yet, this term has been employed differently, and so let me suggest two alternatives. Some of the erroneous forms occur more frequently if the production algorithm is run more quickly: these could be seen as *fast speech forms* in a broad sense. Whereas other forms emerge independently of the production speed, at least in OT and in the strict domination limit of q-HG: these can be identified as *irregular forms*. An example is progressive voice assimilation in some special cases in Dutch, a language which otherwise displays regressive voice assimilation exclusively, "as a rule" (Biró 2006).

The moral is that linguists need not struggle to have their grammars encompass each and every form accepted by the native speaker. It might be a more fruitful strategy to discount some forms, and to aim at a simpler grammar. Then, the irregular forms contravening the general "rules" may simply turn out to be errors made by the grammars that do not count.

Acknowledgements

This research was supported by a *Marie Curie Career Integration Grant* within the *7th EU Framework Programme* (grant no. 631599, "MeMoLI"). The computations were run on the *Atlasz* HPC cluster of the Eötvös Loránd University. The very first draft of this paper grew out from a discussion with Reinhard Blutner in 2007. The author acknowledges the comments of an anonymous reviewer.

Appendix: Numerical results of the computer experiments

Table 4: Precision of the good case grammar (26)

$m =$	10	20	50	100	200	500	1000	2000	5000
$q =$1.1	*0.518*	*0.527*	*0.541*	*0.555*	*0.568*	*0.589*	*0.612*	*0.642*	*0.682*
1.2	*0.534*	*0.550*	*0.578*	*0.601*	*0.627*	*0.665*	*0.705*	*0.753*	*0.815*
1.5	*0.572*	*0.607*	*0.665*	*0.711*	*0.758*	*0.822*	*0.881*	*0.933*	*0.975*
2.0	*0.612*	*0.669*	*0.757*	*0.821*	*0.877*	*0.937*	*0.975*	*0.994*	*0.999*
3.0	*0.659*	*0.739*	*0.850*	*0.918*	*0.962*	*0.991*	*0.999*	*1.000*	*1.000*
5.0	*0.705*	*0.803*	*0.920*	*0.972*	*0.994*	*1.000*	*1.000*	*1.000*	*1.000*
10	*0.751*	*0.860*	*0.965*	*0.994*	*1.000*	*1.000*	*1.000*	*1.000*	*1.000*
20	*0.780*	*0.894*	*0.983*	*0.999*	*1.000*	*1.000*	*1.000*	*1.000*	*1.000*
50	*0.810*	*0.922*	*0.993*	*1.000*	*1.000*	*1.000*	*1.000*	*1.000*	*1.000*
100	*0.826*	*0.934*	*0.995*	*1.000*	*1.000*	*1.000*	*1.000*	*1.000*	*1.000*
200	*0.836*	*0.942*	*0.997*	*1.000*	*1.000*	*1.000*	*1.000*	*1.000*	*1.000*
500	*0.842*	*0.950*	*0.998*	*1.000*	*1.000*	*1.000*	*1.000*	*1.000*	*1.000*
1000	*0.849*	*0.955*	*0.998*	*1.000*	*1.000*	*1.000*	*1.000*	*1.000*	*1.000*

Table 5: Number of iterations in the good case grammar (26)

$m =$	10	20	50	100	200	500	1000	2000	5000
$q =$1.1	*10.98* ±*1.59*	*22.01* ±*2.43*	*56.65* ±*4.59*	*116.55* ±*7.70*	*239.70* ±*13.22*	*614.41* ±*26.14*	*1232.45* ±*43.56*	*2450.88* ±*73.90*	*6046.75* ±*152.49*
1.2	*10.69* ±*1.56*	*21.38* ±*2.39*	*54.96* ±*4.52*	*113.01* ±*7.60*	*232.27* ±*13.13*	*595.35* ±*26.27*	*1194.37* ±*43.96*	*2374.62* ±*74.55*	*5854.86* ±*153.41*
1.5	*10.02* ±*1.51*	*19.95* ±*2.33*	*51.01* ±*4.46*	*104.57* ±*7.58*	*214.35* ±*13.26*	*548.75* ±*27.16*	*1099.99* ±*45.28*	*2184.72* ±*75.08*	*5379.69* ±*148.67*
2.0	*9.31* ±*1.49*	*18.37* ±*2.32*	*46.58* ±*4.50*	*94.94* ±*7.70*	*193.73* ±*13.32*	*494.83* ±*26.78*	*991.84* ±*42.77*	*1970.70* ±*67.90*	*4857.47* ±*132.46*
3.0	*8.50* ±*1.49*	*16.56* ±*2.36*	*41.36* ±*4.58*	*83.56* ±*7.60*	*169.50* ±*12.45*	*432.51* ±*23.34*	*869.48* ±*36.09*	*1732.28* ±*58.21*	*4277.83* ±*116.03*
5.0	*7.71* ±*1.51*	*14.76* ±*2.42*	*36.13* ±*4.53*	*72.24* ±*7.00*	*145.99* ±*10.68*	*373.31* ±*19.48*	*753.75* ±*31.07*	*1505.55* ±*51.08*	*3723.11* ±*102.70*
10	*6.92* ±*1.56*	*12.92* ±*2.48*	*30.77* ±*4.31*	*60.94* ±*6.09*	*122.96* ±*8.87*	*315.16* ±*16.69*	*638.67* ±*27.16*	*1279.13* ±*44.74*	*3168.20* ±*89.88*
20	*6.35* ±*1.61*	*11.55* ±*2.52*	*26.82* ±*4.06*	*52.77* ±*5.41*	*106.34* ±*7.86*	*272.64* ±*14.86*	*553.96* ±*24.46*	*1111.90* ±*40.24*	*2758.20* ±*80.46*
50	*5.76* ±*1.65*	*10.18* ±*2.53*	*23.00* ±*3.78*	*44.91* ±*4.86*	*90.26* ±*7.00*	*231.19* ±*13.14*	*470.90* ±*21.77*	*947.91* ±*35.78*	*2355.86* ±*70.99*
100	*5.43* ±*1.67*	*9.40* ±*2.54*	*20.83* ±*3.65*	*40.42* ±*4.58*	*81.00* ±*6.51*	*207.30* ±*12.12*	*422.72* ±*20.16*	*852.58* ±*33.13*	*2122.12* ±*65.46*
200	*5.18* ±*1.69*	*8.77* ±*2.55*	*19.07* ±*3.54*	*36.77* ±*4.39*	*73.50* ±*6.14*	*187.85* ±*11.30*	*383.30* ±*18.83*	*774.45* ±*30.92*	*1930.45* ±*60.88*
500	*4.91* ±*1.72*	*8.09* ±*2.55*	*17.21* ±*3.44*	*32.89* ±*4.17*	*65.49* ±*5.75*	*167.04* ±*10.38*	*341.04* ±*17.31*	*690.54* ±*28.44*	*1724.52* ±*55.81*
1000	*4.73* ±*1.75*	*7.68* ±*2.55*	*16.05* ±*3.38*	*30.50* ±*4.06*	*60.52* ±*5.50*	*154.14* ±*9.81*	*314.71* ±*16.35*	*638.14* ±*26.91*	*1595.72* ±*52.61*

Table 6: Precision of the bad case grammar (29)

$m =$	10	20	50	100	200	500	1000	2000	5000
$q =$1.1	*0.621*	*0.679*	*0.762*	*0.824*	*0.877*	*0.938*	*0.976*	*0.994*	*0.999*
1.2	*0.611*	*0.666*	*0.745*	*0.806*	*0.858*	*0.922*	*0.966*	*0.990*	*0.999*
1.5	*0.589*	*0.634*	*0.704*	*0.760*	*0.811*	*0.878*	*0.931*	*0.971*	*0.993*
2.0	*0.567*	*0.601*	*0.658*	*0.706*	*0.752*	*0.814*	*0.871*	*0.927*	*0.972*
3.0	*0.544*	*0.568*	*0.608*	*0.642*	*0.678*	*0.728*	*0.778*	*0.836*	*0.903*
5.0	*0.526*	*0.539*	*0.565*	*0.587*	*0.611*	*0.644*	*0.678*	*0.722*	*0.785*
10	*0.513*	*0.518*	*0.532*	*0.542*	*0.555*	*0.574*	*0.591*	*0.614*	*0.653*
20	*0.506*	*0.509*	*0.514*	*0.520*	*0.527*	*0.536*	*0.545*	*0.557*	*0.576*
50	*0.502*	*0.504*	*0.506*	*0.508*	*0.510*	*0.513*	*0.517*	*0.522*	*0.529*
100	*0.500*	*0.502*	*0.502*	*0.504*	*0.505*	*0.507*	*0.508*	*0.511*	*0.514*
200	*0.500*	*0.500*	*0.501*	*0.502*	*0.502*	*0.503*	*0.504*	*0.506*	*0.508*
500	*0.501*	*0.500*	*0.500*	*0.502*	*0.501*	*0.501*	*0.502*	*0.502*	*0.503*
1000	*0.500*	*0.500*	*0.500*	*0.501*	*0.501*	*0.500*	*0.500*	*0.501*	*0.502*

Table 7: Number of iterations in the bad case grammar (29)

$m =$	10	20	50	100	200	500	1000	2000	5000
$q =$1.1	*10.14* ±*1.69*	*20.22* ±*2.68*	*51.71* ±*5.26*	*105.81* ±*9.03*	*216.63* ±*15.74*	*552.39* ±*31.23*	*1103.37* ±*49.89*	*2186.02* ±*80.05*	*5373.92* ±*157.94*
1.2	*9.85* ±*1.64*	*19.61* ±*2.59*	*50.16* ±*5.06*	*102.73* ±*8.70*	*210.45* ±*15.16*	*537.24* ±*30.37*	*1073.91* ±*48.89*	*2128.33* ±*78.65*	*5232.02* ±*154.65*
1.5	*9.16* ±*1.54*	*18.21* ±*2.39*	*46.57* ±*4.59*	*95.50* ±*7.86*	*195.92* ±*13.76*	*501.70* ±*28.10*	*1005.05* ±*46.17*	*1993.59* ±*75.06*	*4901.74* ±*147.26*
2.0	*8.41* ±*1.43*	*16.66* ±*2.19*	*42.58* ±*4.13*	*87.41* ±*7.01*	*179.55* ±*12.19*	*461.50* ±*25.14*	*927.14* ±*42.06*	*1842.21* ±*69.73*	*4533.82* ±*137.79*
3.0	*7.56* ±*1.32*	*14.86* ±*1.98*	*37.90* ±*3.67*	*77.84* ±*6.12*	*160.10* ±*10.56*	*413.03* ±*21.79*	*833.04* ±*36.52*	*1659.85* ±61.44	*4092.47* ±*123.89*
5.0	*6.70* ±*1.21*	*13.07* ±*1.80*	*33.20* ±*3.27*	*68.14* ±*5.39*	*140.28* ±*9.17*	*362.96* ±*18.78*	*734.92* ±*31.33*	*1468.62* ±*52.62*	*3629.50* ±*107.00*
10	*5.83* ±*1.11*	*11.22* ±*1.62*	*28.32* ±*2.90*	*58.05* ±*4.73*	*119.51* ±*7.94*	*309.87* ±*16.23*	*629.85* ±*26.94*	*1262.57* ±*44.85*	*3128.02* ±*90.77*
20	*5.17* ±*1.06*	*9.83* ±*1.50*	*24.63* ±*2.63*	*50.42* ±*4.26*	*103.75* ±*7.10*	*269.27* ±*14.46*	*548.93* ±*24.16*	*1103.20* ±*40.01*	*2738.12* ±*80.23*
50	*4.50* ±*0.96*	*8.45* ±*1.37*	*20.98* ±*2.37*	*42.85* ±*3.80*	*88.11* ±*6.29*	*228.78* ±*12.75*	*467.81* ±*21.47*	*943.04* ±*35.44*	*2345.53* ±*70.64*
100	*4.12* ±*0.88*	*7.65* ±*1.29*	*18.86* ±*2.22*	*38.43* ±*3.52*	*78.99* ±*5.81*	*205.15* ±*11.74*	*420.16* ±*19.87*	*848.85* ±*32.77*	*2114.38* ±*65.13*
200	*3.84* ±*0.84*	*7.00* ±*1.23*	*17.13* ±*2.09*	*34.83* ±*3.30*	*71.53* ±*5.41*	*185.80* ±*10.88*	*381.02* ±*18.51*	*771.42* ±*30.52*	*1924.41* ±*60.45*
500	*3.54* ±*0.86*	*6.30* ±*1.15*	*15.28* ±*1.94*	*30.98* ±*3.05*	*63.56* ±*4.98*	*165.08* ±*9.95*	*338.97* ±*17.01*	*687.90* ±*28.17*	*1719.52* ±*55.43*
1000	*3.33* ±*0.86*	*5.88* ±*1.11*	*14.13* ±*1.85*	*28.58* ±*2.89*	*58.59* ±*4.71*	*152.17* ±*9.38*	*312.68* ±*16.07*	*635.62* ±*26.62*	*1591.19* ±*52.31*

References

Bíró, T. 2003. Quadratic alignment constraints and finite state Optimality Theory. In Proceedings of the Workshop on Finite-State Methods in Natural Language Processing (FSMNLP), held within EACL-03, Budapest. 119–126. Also: ROA-600.

Bíró, T. 2005a. How to define Simulated Annealing for Optimality Theory? In J. Rogers (ed.) Proceedings of FG-MoL 2005: The 10th Conference on Formal Grammar and The 9th Meeting on Mathematics of Language. Edinburgh: Center for the Study of Language and Information. 49–60.

Bíró, T. 2005b. When the hothead speaks: Simulated Annealing Optimality Theory for Dutch fast speech. In T. v. d. Wouden, M. Poß, H. Reckman and C. Cremers (eds.) Computational Linguistics in the Netherlands 2004: Selected papers from the fifteenth CLIN meeting. Utrecht: LOT Publications. 13–28.

Biró, T. 2006. Finding the right words: Implementing Optimality Theory with simulated annealing. Doctoral dissertation. University of Groningen. ROA-896.

Biró, T. 2007. The benefits of errors: Learning an OT grammar with a structured candidate set. In CACLA '07: Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition. Prague: Association for Computational Linguistics. 81–88.

Biró, T. 2009. Elephants and optimality again: SA-OT accounts for pronoun resolution in child language. In B. Plank, E. Tjong Kim Sang and T. Van de Cruys (eds.) Computational Linguistics in the Netherlands 2009. Groningen: LOT, no. 14 in LOT Occasional Series, 9–24.

Biró, T. 2013. Towards a robuster interpretive parsing: Learning from overt forms in Optimality Theory. Journal of Logic, Language and Information 22. 139–172.

Boersma, P. 1997. How we learn variation, optionality, and probability. Proceedings of the Institute of Phonetic Sciences, Amsterdam (IFA) 21. 43–58.

Boersma, P. 2011. A programme for bidirectional phonology and phonetics and their acquisition and evolution. In A. Benz and J. Mattausch (eds.) Bidirectional Optimality Theory. Amsterdam & Philadelphia: John Benjamins. 33–72.

Boersma, P. and B. Hayes. 2001. Empirical tests of the Gradual Learning Algorithm. Linguistic Inquiry 32. 45–86.

Boersma, P. and J. Pater. 2008. Convergence properties of a gradual learning algorithm for Harmonic Grammar. ROA-970.

Černý, V. 1985. Thermodynamical approach to the travelling salesman problem. Journal of Optimization Theory and Applications 45. 41–51.

Chomsky, N. 1965. Aspects of the theory of syntax. Cambridge, MA: MIT Press.

Eisner, J. 1997. Efficient generation in Primitive Optimality Theory. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-1997) and 8th EACL. Madrid, 313–320. Also: ROA-206.

Eisner, J. 2000. Easy and hard constraint ranking in Optimality Theory: Algorithms and complexity. In J. Eisner, L. Karttunen and A. Thériault (eds.) Finite-state phonology: Proc. of the 5th SIGPHON workshop. Luxembourg, 22–33.

Gigerenzer, G., P. M. Todd and the ABC Research Group. 1999. Simple heuristics that make us smart. Oxford: Oxford University Press.

Goldwater, S. and M. Johnson. 2003. Learning OT constraint rankings using a maximum entropy model. In J. Spenader, A. Eriksson and O. Dahl (eds.) Proceedings of the Stockholm workshop on Variation within Optimality Theory. Stockholm: Stockholm University. 111–120.

González, C. 2005. Segmental alternations in Yaminahua. In Proc. 24th West Coast Conference in Formal Linguistics, Simon Fraser University, Vancouver, Canada.

Graf, T. 2017. The power of locality domains in phonology. Phonology 34. 1–21.

Heinz, J., G. M. Kobele and J. Riggle. 2009. Evaluating the complexity of Optimality Theory. Linguistic Inquiry 40. 277–288.

Henderson, D., S. H. Jacobson and A. W. Johnson. 2003. The theory and practice of simulated annealing. In F. Glover and G. A. Kochenberger (eds.) Handbook of metaheuristics. New York: Kluwer. 287–319.

Idsardi, W. J. 2006a. Misplaced optimism. ROA-840.

Idsardi, W. J. 2006b. A simple proof that Optimality Theory is computationally intractable. Linguistic Inquiry 37. 271–275.

Jackendoff, R. 2007. Language, consciousness, culture: Essays on mental structure. Cambridge, MA: MIT Press.

Jäger, G. 2002. Some notes on the formal properties of bidirectional Optimality Theory. Journal of Logic, Language and Information 11. 427–451.

Jäger, G. and A. Rosenbach. 2006. The winner takes it all – almost: Cumulativity in grammatical variation. Linguistics 44. 937–971.

Karttunen, L. 2006. The insufficiency of paper-and-pencil linguistics: The case of Finnish prosody. In R. M. Kaplan, M. Butt, M. Dalrymple and T. H. King (eds.) Intelligent linguistic architectures. Stanford: CSLI. 287–300.

Kirkpatrick, S., C. D. Gelatt Jr. and M. P. Vecchi. 1983. Optimization by simulated annealing. Science 220. 671–680.

Kornai, A. 2006a. Guarded optimalism. ROA-841.

Kornai, A. 2006b. Is OT NP-hard? ROA-838.

Kornai, A. 2008. Mathematical linguistics. London: Springer.

Magri, G. 2016. Error-driven learning in Optimality Theory and Harmonic Grammar: A comparison. Phonology 33. 493–532.

McCarthy, J. J. 2002. Against gradience. ROA-510.

McCarthy, J. J. 2003. OT constraints are categorical. Phonology 20. 75–138.

McCarthy, J. J. 2007. Restraint of analysis. In S. Blaho, P. Bye and M. Krämer (eds.) Freedom of analysis? Berlin & New York: Mouton de Gruyter. 203–231.

McCarthy, J. J. 2010. An introduction to Harmonic Serialism. Language and Linguistics Compass 4. 1001–1018.

McCarthy, J. J. and A. Prince. 1995. Faithfulness and reduplicative identity. In J. Beckman, L. W. Dickey and S. Urbanczyk (eds.) Papers in Optimailty Theory (University of Massachussetts Occasional Papers in Linguistics 18). Amherst MA: Graduate Linguistic Student Association. 249–384.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller. 1953. Equation of state calculation by fast computing machines. Journal of Chemical Physics 21. 1087–1092.

Newmeyer, F. 1983. Grammatical theory: Its limits and its possibilities. Chicago: University of Chicago Press.

Pater, J. 2009. Weighted constraints in generative linguistics. Cognitive Science 33. 999–1035.

Prince, A. 2002. Anything goes. ROA-536.

Prince, A. and P. Smolensky. 1993. Optimality theory: Constraint interaction in generative grammar. Technical Report TR-2, Center for Cognitive Science, Rutgers University, New Brunswick, N.J. and Technical Report CU-CS-697-93, Department of Computer Science, University of Colorado, Boulder.

Prince, A. and P. Smolensky. 2004. Optimality Theory: Constraint interaction in Generative Grammar. Malden, MA & Oxford: Blackwell.

Pulleyblank, D. and W. J. Turkel. 2000. Learning phonology: Genetic algorithms and Yoruba tongue-root harmony. In J. Dekkers, F. van der Leeuw and J. van De Weijer (eds.) Optimality Theory: Phonology, syntax, and acquisition. Oxford: Oxford University Press. 554–591.

Reeves, C. R. (ed.). 1995. Modern heuristic techniques for combinatorial problems. London: McGraw-Hill.

Smolensky, P. and G. Legendre (eds.). 2006. The harmonic mind: From neural computation to Optimality-Theoretic grammar. Cambridge, MA: MIT Press.

Soderstrom, M., D. W. Mathis and P. Smolensky. 2006. Abstract genomic encoding of universal grammar in Optimality Theory. In Smolensky & Legendre (2006, 403–471).

Tesar, B. and P. Smolensky. 2000. Learnability in Optimality Theory. Cambridge, MA: MIT Press.

Turkel, B. 1994. The acquisition of Optimality Theoretic systems. ROA-11.

Watanabe, S. 2009. Cultural and educational contributions to recent phonological changes in Japanese. Doctoral dissertation. University of Arizona.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 141–152

Language modeling with matrix embeddings

GÁBOR BORBÉLY
Budapest University of Technology and Economics
borbely@math.bme.hu

KEYWORDS

language modeling
neural language model
vector embedding
matrix embedding
matrix algebra

ABSTRACT

Vector representations of words, after decades of being at the periphery of computer linguistics, are today widely used and researched. According to our terminology, representing a word involves a function assigning a vector to each word from a finite set (vocabulary). In this paper we investigate certain properties and limitations of word vectors with the aim of improving them. We also present a novel method for learning not vector, but matrix representation of words. The matrices are the result of gradient descent learning where the objective function rewards the presence of a word in its neighboring context, similar to language modeling.

1. Preliminaries

Vector representations of words, after decades of being at the periphery of computer linguistics, are today widely used and researched.

According to our terminology, *representing a word* involves a function which assigns a vector (in $\mathbb{R}^d$) to every word from a finite set (vocabulary) V.

$$v : V \mapsto \mathbb{R}^d$$

Early experiments focused on language modeling with neural architectures (Xu & Rudnicky 2000 and Bengio et al. 2003) instead of n-gram models (Kneser & Ney 1995). Referred to as *distributional vector semantics*, *word embeddings*, or *word vectors*, they are now off-the-shelf tools in the field of natural language processing (NLP) as of Mikolov et al. (2013a) and Pennington et al. (2014). In case of these tools the function v is learned from a corpus of monolingual, unlabeled, tokenized text. Their learning objective is similar to that of language modeling in the sense that they maximize the likelihood of a word in its neighboring context.

Word vectors have proven useful in several applications: e.g., sentiment analysis Socher et al. (2013), diachronic semantics change Hamilton et al. (2016), zero-shot learning Dinu et al. (2015), neural dependency parsing Dozat et al. (2017), and have also been scientifically investigated, e.g., in Arora et al. (2016).

In this paper we investigate certain properties and limitations of word vectors with the aim of improving them. We also present a novel method for learning not vector, but matrix representations of words.

In sections 2 and 3 we provide some theoretical background. Section 4 presents the actual training objectives and models. Some numerical results are provided in section 5.

2. Vector space structure

As seen in Mikolov et al. (2013b) and in Mikolov et al. (2013c) the linear structure of trained vector models is undeniable, meaning that the semantic structure is well represented by vector operations (linear combination and dot product). From analogy questions (*king-man+woman=queen*) through word similarity (angle of word vectors) and translation ($\underline{v}_{\text{dog}} \cdot \underline{\underline{T}}_{\text{eng to ger}} = \underline{v}_{\text{Hund}}$) to even some phrases (*Chinese+river=Yangtze*), linear vector space structure seems to be empirically justified.

However, word vectors alone are not suitable for composing phrases or sequences of words. In the example above *Chinese+river* is not the same as 'Chinese river', at least not more so than 'river Chinese'. Vector addition is *commutative*, i.e. results are independent of the order of the operands. This is why the vector addition itself is not suited for modeling composition. The sum of words may be used to represent a phrase, but tackling compositionality in general is a demanding task.

In Socher et al. (2013) parse trees were used to recursively process phrases to create sentence representations. In Hill et al. (2016) the LSTM architecture (Long Short-Term Memory, Hochreiter & Schmidhuber 1997) was used to represent phrases. Learning compositional mechanisms to embed entities of various length (words, phrases and sentences) are of central interest to modern neural language processing.

3. Algebras

The question arises naturally: which are the appropriate mathematical structures (and composition rules) for a word embedding. The performance of vector models suggests that vector space structure is a good starting point. We also mentioned that beside the useful + operation, words tend to require an additional operation, which composes them, and this composition is non-commutative. An *algebra* over the real field is a reasonable choice Rudolph & Giesbrecht (2010).

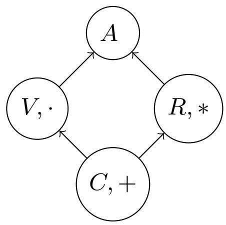


Figure 1: Algebras versus other structures

In Figure 1 the symbols C, V, R and A represent commutative groups, vector spaces, rings, and algebras, respectively. Commutative groups have a commutative addition operator (among others), vector spaces also have dot product (and also scalar multiplication). A ring has addition and (non-commutative) multiplication and an algebra is equipped with all of these operations.

Note that in a group, in theory, one can compute elements like *Chinese*+*river* or even subtract: *Volga* − *Russia*, but there is no way of comparing the result to existing vocabulary entries. In a vector space, the dot product can be used to measure similarities between elements, while scalar multiplication allows us to calculate *averages* over certain elements. In a ring, one can use the multiplication operator to model composition, but it still lacks some properties of the vector space. Algebras meet all of these requirements.

As a special case of algebras, *matrix algebras* consist of square matrices, which are central to our investigations. Hence the name *matrix embedding*: we want to train square matrices for each word in a vocabulary, given a corpus of sentences.

The algebra operations would look something like this:

$$\text{green} + \text{orange} \approx \text{yellow-ish color or a team with these colors}$$
$$\text{green} * \text{orange} \approx \text{“green orange” like an unriped fruit}$$

4. Learning matrices

Let V be our vocabulary: a finite set of symbols (words). Let $\mathcal{C} \subset V^*$ be a collection of sentences, i.e., a corpus. We seek a map which assigns a matrix to each word: $M : V \mapsto \mathbb{R}^{d \times d}$. The size of matrices ($d$) is a model parameter.

In order to train such a map we must impose an objective function that measures how good a sentence is.

$$f : V^* \mapsto \mathbb{R}$$

We wish to find an appropriate f and optimize it with respect to M given the corpus of sentences.

First we make some restrictions on the function f. Since we want to model composition via matrix multiplication, f will be evaluated solely on matrices, not on series of matrices. The score of a sentence should be the score of the product of its words.

$$f(\text{“the dog barks”}) = f(M_{\text{the}} \cdot M_{\text{dog}} \cdot M_{\text{barks}})$$

Note that compositionality takes place in the matrix product, the product of three matrices is also a matrix, which is in the same vector space as its components, although not necessarily in the vocabulary.

As in Pennington et al. (2014), we choose the scoring function to be linear in its components. In the example above, it is linear in all of its inputs: “the”, “dog” and “barks”.

$$f(M_{\text{the}} \cdot M_{\text{dog}} \cdot M_{\text{barks}}) \approx \log \mathbb{P}(\text{“the dog barks”})$$

In our work we choose f in a way similar to Rudolph & Giesbrecht (2010):

$$f(M) = \underline{v}^\top \cdot M \cdot \underline{w}$$
$$f(M_{\text{the}} \cdot M_{\text{dog}} \cdot M_{\text{barks}}) = \underline{v}^\top \cdot M_{\text{the}} \cdot M_{\text{dog}} \cdot M_{\text{barks}} \cdot \underline{w}$$

where $\underline{v}$ and $\underline{w}$ are column vectors, depending on the model which will be specified later.

In the following subsections we introduce various models which implement the above ideas. All of them are suitable for optimization and indeed train the embedding M but with different approaches. Numerical results are presented in section 5.

4.1. Neural network model

The following model one does not make predictions about the probabilities of full sentences, but only about probabilities of individual words appearing in a given context. f shall be such that

$$f(\text{context}_{\text{before}}, \text{word}, \text{context}_{\text{after}}) = \mathbb{P}(\text{word}|\text{context})$$
$$\sum_{w \in V} f(\text{context}_{\text{before}}, w, \text{context}_{\text{after}}) = 1.$$

Our architecture consists of an embedding layer M, a composition layer using matrix dot product, and a readout layer that is a softmax function over the vocabulary (given a fixed context).

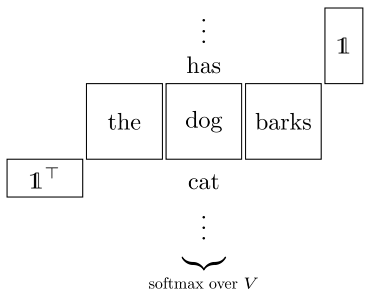


Figure 2: Neural architecture of the matrix embedding model

In formulas, the objective is to minimize the entropy in every context, like:

$$-\log \frac{\exp\left(\mathbb{1}^\top \cdot M_{\text{the}} \cdot M_{\text{dog}} \cdot M_{\text{barks}} \cdot \mathbb{1}\right)}{\sum_{v \in V} \exp\left(\mathbb{1}^\top \cdot M_{\text{the}} \cdot M_v \cdot M_{\text{barks}} \cdot \mathbb{1}\right)} \to \min$$

Or, more precisely, to maximize the following in M.

$$\sum_{(c_b,w,c_a)\in\mathcal{C}} \left[\left(\mathbb{1}^\top \cdot \left(\prod_{b\in c_b} M_b \right) \cdot M_w \cdot \left(\prod_{a\in c_a} M_a \right) \cdot \mathbb{1} \right) - \log \sum_{v\in V} \exp \left(\mathbb{1}^\top \cdot \left(\prod_{b\in c_b} M_b \right) \cdot M_v \cdot \left(\prod_{a\in c_a} M_a \right) \cdot \mathbb{1} \right) \right]$$

where c_b and c_a are the context before and after the word w and the products are ordered (non-commutative) matrix dot products. Contexts are not required to be symmetric or of constant width, an empty product yields the *identity matrix*, which can be used as a placeholder.

Note that this model does not assign probabilities to a whole sentence, only to certain choices of words. The probability of a sentence is hard to measure (see Kornai 2010), therefore we do not require the model to calculate them.

4.2. Direct probabilistic model

We modify the above model in a way that the probability of every sentence, phrase, and word is calculated directly. To obtain a full probabilistic model we eliminate the softmax in the neural model, this is achieved by constraining the matrices M_w to ensure they have a total probability of 1. Then the probability of the skip-gram "the »anything« barks" in the corpus can be defined as

$$\sum_{w\in V} \mathbb{P}(\text{the } w \text{ barks}) = \sum_{w\in V} \mathbb{1}^\top \cdot M_{\text{the}} \cdot M_w \cdot M_{\text{barks}} \cdot \mathbb{1} = \mathbb{P}(\text{the »anything« barks})$$

Since the above formula is linear in the middle matrix, we can calculate a placeholder

$$M_* := \sum_{w\in V} M_w$$

which does not change the probability of any sequence, no matter where it is inserted. We also require that $\mathbb{P}(\text{»anything«}) = 1$.

In this model matrices have an inevitable probabilistic interpretation. We postulate the following *constraints* over M:

- positivity of the elements: $(M_w)_{i,j} \geq 0$,
- right-stochastic sum:

$$\sum_{w \in V} M_w \text{ is a right-stochastic matrix}$$

  i.e., its rows sum up to 1.

Under these conditions we can state the following:

- $\frac{1}{d} \cdot \mathbb{1}^\top \cdot (M_*)^n \cdot \mathbb{1} = 1$ for $n = 0, 1, 2 \ldots$
- If $\underline{v} \in \mathbb{R}^{1 \times d}$ has non-negative entries and sums up to 1, then $\underline{v} \cdot M_*$ also has non-negative entries and sums up to 1 (i.e., keeps the probabilistic row vectors).
- $(M_*)^n$ is also a right-stochastic matrix, therefore

$$\sum_{w_1 \in V} \ldots \sum_{w_n \in V} \frac{1}{d} \mathbb{1}^\top M_{w_1} \cdots M_{w_n} \mathbb{1} = \frac{1}{d} \mathbb{1}^\top (M_*)^n \mathbb{1} = 1.$$

In this setup we can simply calculate the probability of any phrase or series of words as

$$\mathbb{P}(w_1 w_2 \ldots w_n) = \frac{1}{d} \mathbb{1}^\top M_{w_1} M_{w_2} \cdots M_{w_n} \mathbb{1}.$$

This model can be trained on a weighted corpus, where every sentence has an empirical probability p, in this case we must minimize the KL divergence.

$$\underset{\text{constraints on } M}{\arg\min} \sum_{\substack{c \in \mathcal{C} \\ \mathbb{P}(c) = p}} p \cdot \log \left(\frac{p}{\frac{1}{d} \mathbb{1}^\top \left(\prod_{w \in c} M_w \right) \mathbb{1}} \right)$$

If the corpus has no weights then we assume $p \equiv 1$.

The models so far were discriminative models.

4.3. Continuous WFSA

We can generalize weighted finite state automata by modifying the above model, and we can optimize a continuous finite state automaton to fit a weighted language. Similar connections between WFSAs and matrix representations of words can be found in Asaadi & Rudolph (2016). We also introduce a learning algorithm to obtain our matrix embeddings, which in turn can help us learn automata. As future work, these techniques may be relevant in MDL (Minimum Description Length) learning of automata, as in Kornai et al. (2013).

In the previous model the left-hand-side of the product can be considered as a context or state vector.

$$\underbrace{\underbrace{\underbrace{\frac{1}{d}\mathbb{1}^\top \cdot M_{\text{the}} \cdot M_{\text{dog}}}_{\text{previous state}} \cdot M_{\text{barks}}}_{\text{the state after "barks"}} \cdot \mathbb{1}}_{\text{probability of the whole outcome}}$$

In a way, the initial row vector $\frac{1}{d}\mathbb{1}^\top$ is carried through the sentence and we can obtain the probability of the current state by applying the column vector $\mathbb{1}$.

Some modification is needed to justify this intuition and introduce WFSA. We change the constraints on the embedding M, since the state of an automaton should always sum up to 1. In the previous model the sum of the earlier mentioned row vector decreases as the sentence spans. Let M_w be a right-stochastic matrix for every word $w \in V$. Then the automaton starts from the uniform state $\frac{1}{d}\mathbb{1}^\top$ and every word acts as a transition on this state.

$$\underbrace{\underline{v}}_{\text{state}} \overbrace{\rightarrow}^{\text{action of } w} \underbrace{\underline{v} \cdot M_w}_{\text{new state}} \tag{1}$$

Some additional action is required, since the sum of every state is now 1 and we want to obtain meaningful probabilities. Let $R \in \mathbb{R}^{d \times |V|}$ be a matrix of non-negative entries which is responsible for emissions. In neural network terminology we would call this the *readout layer.*

At every state $\underline{v}$ the columns of the matrix R determine which word should follow.

$$\mathbb{P}(\text{next word is } w \mid \text{ state } \underline{v}) = \underline{v} \cdot \underbrace{R_{\bullet,w}}_{w^{\text{th}} \text{ column}} \tag{2}$$

Constraints on R and M are listed below.

- M_w is a right-stochastic matrix $\forall w \in V$.
- The rows of R sum up to 1 (and R has non-negative entries).

Under these constraints an automaton arises:

- The states are $1, 2 \dots d$.
- The initial state is uniform over the states: $\frac{1}{d}\mathbb{1}^\top$.
- A word w acts as a transition function on the states as in (1).
- The outcomes (or emissions) at a given state (probabilistic row vector) $\underline{v}$ follow as in (2).

Finally, the probability of an emission sequence is the following product:

$$\mathbb{P}(w_1 w_2 \dots w_n) = \underbrace{\mathbb{1}^\top R_{\bullet,w_1}}_{\mathbb{P}(w_1)} \cdot \underbrace{\mathbb{1}^\top M_{w_1} R_{\bullet,w_2}}_{\mathbb{P}(w_2|w_1)} \cdots \underbrace{\mathbb{1}^\top \prod_{i=1}^{n-1} M_{w_i} R_{\bullet,w_n}}_{\mathbb{P}(w_n|w_1 w_2 \dots w_{n-1})}$$

Given a corpus or a weighted language, we can use the same objective function as in the previous section and train M and R.

Note that, unlike in the previous two models, this model is not sensitive to future words. The next emission and state does not depend on following words. In contrast to the previous one, this is a generative model.

5. Results

Our experimental setup used the UMBC gigaword corpus (Han et al. 2013) which was tokenized and split at sentence boundaries (punctuation part-of-speech tag). The words were not converted into lowercase. It contains about 3.338G words, the average length of a sentence is about 24 with standard deviation 15. For computational reasons, we excluded long sentences, leaving 126.7M sentences to work with.

Words with frequency below 52 were replaced with a unique symbol `<UNK>`, leaving roughly 100k types in the vocabulary (precisely 100147).

The implementation is not detailed herein, but the code is available.[1]

We encountered some serious numerical obstacles in case of model 4.1. We are not certain whether these numerical issues are caused by the

[1] https://github.com/hlt-bme-hu/lm_me, see C++ code for neural model, python (theano) implementation for the other two models.

implementation or by the mathematical model, but the problem occurs if the stochastic gradient descent encounters the same token several times in the same sentence. Nevertheless, this problem did not occur in models 4.2 and 4.3. The first model differs from the others in several aspects: implementation language, mathematical model, and also in gradient descent strategy.

The performance of each model was measured on Google Analogy questions Mikolov et al. (2013a), see evaluation code below.[2] Cosine similarity was used on the flattened matrices. The third model did not achieve meaningful quality within reasonable computation time, here we only present results of the second model.

The table below shows the number of correctly answered questions of each trained model. *Commutative* means that the matrices were 100×100 diagonal matrices; they form a commutative algebra. This can be considered as a fallback to word vectors. The *dense* models consist of 10×10 dense matrices.

Algebra	Model	Nr. correct
commutative	4.2	555
dense	4.2	81

The model 4.3 could answer 1 or 2 questions after equal amount of training.

6. Outlook

We introduced several techniques to train matrix embeddings of words with various numerical efficiency and quality.

Training high quality embeddings and/or automata is our future interest. There are some obvious obstacles in computation time, since the training of a well tuned embedding usually takes days and matrix models are expected to require even more computation time.

A possible computational enhancement is the use of structured, sparse matrices, of which we train only certain elements, hence taking a sub-algebra of the full matrix algebra. This hastens some calculations but keeps the desired algebra properties intact. To this end, further studies of matrix algebras and their sub-algebras are considered.

[2] https://github.com/hlt-bme-hu/eval-embed

Currently these experiments are in a preliminary state, but many improvements and applications are possible. As my supervisor, András Kornai, has once described it, "Like socialism; appealing idea, but not working in practice".

References

Arora, S., Y. Li, Y. Liang, T. Ma and A. Risteski. 2016. RAND-WALK: A latent variable model approach to word embeddings. Transactions of the Association for Computational Linguistics 4. 385–399.

Asaadi, S. and S. Rudolph. 2016. On the correspondence between Compositional Matrix-Space Models of language and weighted automata. In Proceedings of the ACL Workshop on Statistical Natural Language Processing and Weighted Automata (StatFSM 2016).

Bengio, Y., R. Ducharme, P. Vincent and C. Janvin. 2003. A neural probabilistic language model. Journal of Machine Learning Research 3. 1137–1155.

Dinu, G., A. Lazaridou and M. Baroni. 2015. Improving zero-shot learning by mitigating the hubness problem. ICLR 2015, Workshop Track.

Dozat, T., P. Qi and C. D. Manning. 2017. Stanford's Graph-based Neural Dependency Parser at the CoNLL 2017 Shared Task. In Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. Vancouver, Canada: Association for Computational Linguistics, 20–30.

Hamilton, W. L., J. Leskovec and D. Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 1489–1501.

Han, L., A. L. Kashyap, T. Finin, J. Mayfield and J. Weese. 2013. UMBC_EBIQUITY-CORE: Semantic textual similarity systems. In Second Joint Conference on Lexical and Computational Semantics (*SEM). 44–52.

Hill, F., K. Cho, A. Korhonen and Y. Bengio. 2016. Learning to understand phrases by embedding the dictionary. Transactions of the Association for Computational Linguistics 4. 17–30.

Hochreiter, S. and J. Schmidhuber. 1997. Long short-term memory. Neural Computation 9. 1735–1780.

Kneser, R. and H. Ney. 1995. Improved backing-off for m-gram language modeling. In International Conference on Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., vol. 1. IEEE, vol. 1, 181–184.

Kornai, A. 2010. Rekurzívak-e a természetes nyelvek? [Are natural languages recursive?] Magyar Tudomány 171. 994–1005.

Kornai, A., A. Zséder and G. Recski. 2013. Structure learning in weighted languages. In Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13). Sofia, Bulgaria, 72–82.

Mikolov, T., K. Chen, G. Corrado and J. Dean. 2013a. Efficient estimation of word representations in vector space. In Y. Bengio and Y. LeCun (eds.) Workshop proceedings, International Conference on Learning Representations (ICLR 2013).

Mikolov, T., Q. V. Le and I. Sutskever. 2013b. Exploiting similarities among languages for machine translation. Manuscript. https://arxiv.org/abs/1309.4168.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado and J. Dean. 2013c. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani and K. Q. Weinberger (eds.) Advances in neural information processing systems 26. Red Hook, NY: Curran Associates. 3111–3119.

Pennington, J., R. Socher and C. Manning. 2014. Glove: Global vectors for word representation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014).

Rudolph, S. and E. Giesbrecht. 2010. Compositional matrix-space models of language. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Uppsala, Sweden, 907–916.

Socher, R., J. Bauer, C. D. Manning and A. Y. Ng. 2013. Parsing with compositional vector grammars. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013). Sofia, Bulgaria: Association for Computational Linguistics, 455–465.

Xu, W. and A. I. Rudnicky. 2000. Can artificial neural networks learn language models? In International Conference on Statistical Language Processing. Beijing, 202–205.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 153–185

Dataflow matrix machines and V-values: A bridge between programs and neural nets

MICHAEL BUKATIN
HERE North America LLC
bukatin@cs.brandeis.edu

JON ANTHONY
Boston College
jsa.aerial@gmail.com

KEYWORDS

dataflow matrix machines
linear streams
self-referential neural networks
continuous deformations of programs
V-values
variadic neurons

ABSTRACT

Dataflow matrix machines (DMMs) generalize neural nets by replacing streams of numbers with linear streams (streams supporting linear combinations), allowing arbitrary input and output arities for activation functions, countable-sized networks with finite dynamically changeable active part capable of unbounded growth, and a very expressive self-referential mechanism.

DMMs are suitable for general-purpose programming, while retaining the key property of recurrent neural networks: programs are expressed via matrices of real numbers, and continuous changes to those matrices produce arbitrarily small variations in the associated programs.

Spaces of V-values (vector-like elements based on nested maps) are particularly useful, enabling DMMs with variadic activation functions and conveniently representing conventional data structures.

1. Introduction

András Kornai wrote his *Mathematical Linguistics* book (Kornai 2008) while he and the first author of the present paper were working in the same office at MetaCarta during the previous decade and sharing many fruitful moments. The book published ten years ago was written as an introduction to the mathematical foundations of linguistics for computer scientists, engineers, and mathematicians interested in natural language processing.

Dataflow matrix machines emerged two years ago (see Bukatin & Matthews 2015b) and a series of technical papers have been written on the subject since then (Bukatin et al. 2016a;b;c;d; Bukatin & Anthony

2017). The present paper is meant to be an introduction to the subject for researchers and engineers working in other fields. We tried to keep the style of chapters and sections of *Mathematical Linguistics* in mind while writing this article.

* * *

Artificial neural networks are a powerful machine learning platform based on processing the streams of numbers.

It is long known that recurrent neural networks are expressive enough to encode any algorithm, if they are equipped with a reasonable form of unbounded memory (McCulloch & Pitts 1943; Pollack 1987; Siegelmann & Sontag 1995). There is a long history of synthesis of algorithms expressed as neural networks both by compilation and by machine learning methods.

However, conventional neural networks belong to the class of *esoteric programming languages* and do not constitute a convenient platform for manual software engineering.

In particular, there is a considerable history of using neural networks to synthesize and modify other neural networks, including self-modification. However, the limitations of conventional neural networks as a software engineering platform make efforts of this kind quite challenging.

The main key point of the approach of *dataflow matrix machines* is that the natural degree of generality for the neural model of computations is not the streams of numbers, but arbitrary *linear streams*.

The other enhancements dataflow matrix machines make to the neural model of computations are neurons of *variable input and output arity*, novel models of unbounded memory based on countable-sized weight-connectivity matrices with finite number of non-zero weights at any given time and, more generally, on streams of countable-dimensional vectors, and *explicit self-referential facilities*.

This results in a more powerful and expressive machine learning platform.

When one considers dataflow matrix machines as a software engineering framework, it turns out that the restriction to linear streams and to programs which admit continuous deformations is less severe than one could have thought a priori given the discrete nature of conventional programming languages.

Dataflow matrix machines are considerably closer to being a general-purpose programming platform than recurrent neural nets, while retaining the key property of recurrent neural nets that large classes of programs can be parametrized by matrices of numbers, and therefore synthesizing appropriate matrices is sufficient to synthesize programs.

Linear streams. Dataflow matrix machines are built around the notion of linear streams. Generally speaking, we say that a space of streams is a *space of linear streams*, if a meaningful notion of linear combination of several streams with numerical coefficients is well-defined.

The simplest example of a space of linear streams is the space of sequences of numbers. A slightly more complicated example comes from considering a vector space V and the space of sequences of its elements, $(v_1, v_2, \ldots)$.

In the first few sections of this paper, this simple version of the notion of linear streams would be sufficient. The discrete time is represented by non-negative integers, a particular vector space V is fixed, and the space of functions from time to V forms the space of linear streams in question.

To distinguish between streams based on different linear spaces, e.g., V_1 and V_2, we talk about different *kinds of linear streams*.

In section 5, we describe a sufficiently general notion of linear streams which includes, for example, streams of samples from sequences of probability distributions over an arbitrary measurable space X. All constructions in the present paper work in this degree of generality.

Neuron types. The originally developed formalism of dataflow matrix machines was heavily typed (Bukatin et al. 2016b). One considered a diverse collection of *kinds of linear streams*, and a diverse collection of *types of neurons* with explicit fixed input and output arities.

The more recent version is close to being type free. It uses a single kind of linear streams based on a "sufficiently universal" space of elements we call V-values. V-values enable the use of *variadic neurons* which have arbitrary input and output arities, eliminating the need to keep track of input and output arities. The neuron types still exist in that they have different activation functions.

Structure of this paper. We start with an informal introduction to recurrent neural networks and to the typed version of dataflow matrix machines (DMMs) in section 2. In section 3, we present the theory of V-values. In section 4, we describe DMMs based on V-values and variadic neurons. We discuss linear streams in section 5. In that section we also discuss embeddings of discrete objects into vector spaces, and into spaces of linear streams.

Programming patterns in DMMs are presented in section 6. A very expressive self-referential mechanism which is a key element of our approach is presented in section 7.

The issues related to expressing the network topology are the subject of section 8 and the issues related to subnetworks and modularization are the subject of section 9.

We discuss some of the potential approaches to using DMMs in machine learning in section 10. The concluding section 11 contains historical remarks and discussion of related work.

2. From recurrent neural networks to dataflow matrix machines

The essence of artificial neural architectures is that linear and nonlinear transformations are interleaved. Then one can control neural computations by only modifying the linear part and keeping the non-linear part fixed.

Therefore, neural architectures such as *recurrent neural networks* (RNNs) can be viewed as "two-stroke engines" (Figure 1), where the "two-stroke cycle" of a linear "down movement" followed by a typically non-linear "up movement" is repeated indefinitely.

The network consists of the weight matrix **W** and the neurons. The neuron k has input and output streams of numbers, x_k^t and y_k^t, associated with it, where t is discrete time. The network also has streams of numbers i_m^t representing external inputs, and streams of numbers o_n^t representing external outputs.

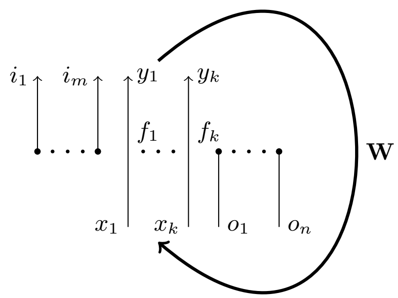


Figure 1: "Two-stroke engine" for an RNN. Figure from Bukatin & Anthony (2017).

On the "down movement", neuron inputs and network external outputs are computed by applying linear transformation $\mathbf{W}$ to the neuron outputs and network external inputs: $(x_1^{t+1}, \ldots, x_k^{t+1}, o_1^{t+1}, \ldots, o_n^{t+1})^\top = \mathbf{W} \cdot (y_1^t, \ldots, y_k^t, i_1^t, \ldots, i_m^t)^\top$. On the "up movement", the neurons calculate their outputs from their inputs using activation functions f_k which are built into each neuron k and are usually non-linear: $y_1^{t+1} = f_1(x_1^{t+1}), \ldots, y_k^{t+1} = f_k(x_k^{t+1})$.

Note that the computations during the "up movement" are local to the neuron in question, while the computations during the linear "down movement" are potentially quite global, as any neuron output might potentially be linked to any neuron input by a non-zero element of $\mathbf{W}$.

Now, moving from RNNs to *dataflow matrix machines* (DMMs), consider a finite or countable collection of *kinds of linear streams*, a finite or countable collection of *neuron types*, with every neuron type specifying non-negative integer number of inputs, non-negative integer number of outputs, the kind of linear streams associated with each input and each output, and an activation function transforming the inputs to the outputs.

Take a countable collection of neurons of each type, so that we have a countable-sized overall network. However, we'll make sure that only a finite part of this network is active at any given time (similarly to only a finite part of the Turing machine tape having non-blank symbols at any given time), and that processing time and memory are only spent on working with the currently active part, while the rest exists simply as potentially infinite address space.

The network consists of a countable-sized connectivity matrix $\mathbf{W}$ and a countable-sized collection of neurons described in the previous paragraph. The connectivity matrix $\mathbf{W}$ depends[1] on discrete time t.

The matrix element $w^t_{(i,C_k),(j,C_l)}$ is the weight linking the output j of the neuron C_l to the input i of the neuron C_k at the moment t. We impose the condition that at any given moment of time t only finite number of matrix elements $w^t_{(i,C_k),(j,C_l)}$ are non-zero. Hence the connectivity matrix is inherently sparse, and the structure of its non-zero weights determines the actual connectivity pattern of the network at any given moment of time.

The DMM "two-stroke engine" in a fashion similar to that of RNNs has a "two-stroke cycle" consisting of a linear "down movement" followed by an "up movement" performed by the activation functions of neurons (Figure 2). This "two-stroke cycle" is repeated indefinitely.

[1] When $\mathbf{W}$ changes with time, this change can be controlled from the outside or by the network itself via the self-referential mechanism described in section 7.

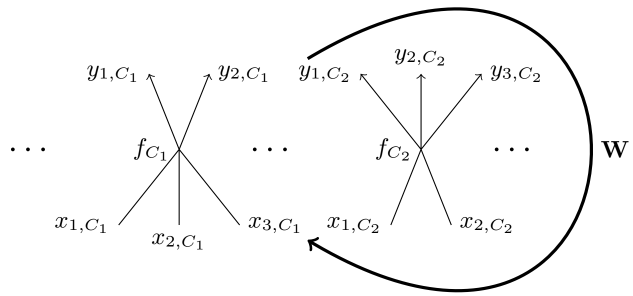


Figure 2: "Two-stroke engine" for a standard DMM (Bukatin & Anthony 2017). Two of its neurons, C_1 and C_2, are explicitly shown.

"Down movement" is defined as follows. For all inputs x_{i,C_k} where there is a non-zero weight $w^t_{(i,C_k),(j,C_l)}$:

$$x^{t+1}_{i,C_k} = \sum_{\{(j,C_l)|w^t_{(i,C_k),(j,C_l)} \neq 0\}} w^t_{(i,C_k),(j,C_l)} * y^t_{j,C_l}. \tag{1}$$

Note that x^{t+1}_{i,C_k} and y^t_{j,C_l} may no longer be numbers, but elements of linear streams x_{i,C_k} and y_{j,C_l}, so in order for Equation 1 to be well-defined we impose the type correctness condition which states that $w^t_{(i,C_k),(j,C_l)}$ is allowed to be non-zero only if x_{i,C_k} and y_{j,C_l} belong to the same *kind* of linear streams.[2]

We call a neuron C active at the time t, if there is at least one non-zero connectivity weight from $\mathbf{W}$ associated with one of its inputs or outputs. Since $\mathbf{W}$ has only a finite number of non-zero weights at any given time, there are only a finite number of active neurons in the network at any given time.

"Up movement" is defined as follows. For all active neurons C:

$$y^{t+1}_{1,C}, ..., y^{t+1}_{n,C} = f_C(x^{t+1}_{1,C}, ..., x^{t+1}_{m,C}). \tag{2}$$

Here m is the input arity of neuron C and n is its output arity, so f_C has m inputs and n outputs. If $m = 0$, then f_C has no arguments. If $n = 0$, then the neuron just consumes data, and does not produce streams on

[2] Recall that the number of inputs and outputs of a neuron C, the kind of linear streams associated with each particular input or output of this neuron, and the built-in activation function f_C of this neuron are determined by the type of the neuron in question.

the "up movement". Given that the input and output arities of neurons are allowed to be zero, special handling of network inputs and outputs which was necessary for RNNs (Figure 1) is not needed here. The neurons responsible for network input and output are included on par with all other neurons.

The resulting formalism is very powerful, and we discuss what can be done with it later in the paper. However, its complexity is a bit unpleasant. The need to keep track of various kinds of linear streams and of the details of various neuron types is rather tiresome. It would be great to have only one sufficiently expressive kind of linear streams, and, moreover, to avoid the need to specify the arity of activation functions, while still enjoying the power of having multiple inputs and outputs within a single neuron. The spaces of *V-values* discussed in the next section allow us to achieve just that, while further increasing the power of a single neuron.

3. V-values

In this section, we define vector space V which is sufficiently rich to represent vectors from many other spaces encountered in practice.

In section 5.3, we show how to enrich this construction in those situations where it is not sufficiently universal, resulting in a family of vector spaces.

We call both the elements of these vector spaces and the hash-map-based representations of those elements *V-values*. In our context, V-values play the role somewhat similar to the role of S-expressions in the context of Lisp.

Implementation-wise, we create a version of S-expressions which is dictionary-based, rather than list-based. In this section, we require all atoms of those dictionary-based S-expressions to be numbers. A more general form of leaves in V-values is considered in section 5.3.

Speaking more formally, we start with a finite or countable alphabet L of labels (which we sometimes call tokens or keys). One can think about elements of L as words from some language defined over some other alphabet, which allows us to think about meaningful languages of labels.

We are going to consider several equivalent ways to define tree-like structures with intermediate nodes labeled by elements of L and with leaves labeled by numbers. Some of these ways are "depth-first", and they are easier to present mathematically, and some are "breadth-first", and they are more fundamental to us, as we use them in our implementation and as they enable the use of variadic neurons.

These tree-like structures can be viewed as

– Finite linear combinations of finite strings;
– Finite prefix trees with numerical leaves;
– Sparse "tensors of mixed rank" with finite number of non-zero elements;
– Recurrent maps from $V \cong \mathbb{R} \oplus (L \to V)$ admitting finite descriptions.

3.1. Finite linear combinations of finite strings

To start with "depth-first" methods, consider space L^* of finite sequences of elements of L (including the empty sequence), and construct V as the vector space of formal finite linear combinations[3] of the elements of L^* over $\mathbb{R}$.

We denote the empty sequence of elements of L as ε, and we denote non-empty sequences of elements of L, $(l_1, \dots, l_n)$, as $l_1 \rightsquigarrow \dots \rightsquigarrow l_n$. Since we are talking about formal finite linear combinations of elements of L^*, we need a notation for the multiplication of real number α and generator $l_1 \rightsquigarrow \dots \rightsquigarrow l_n$, $\alpha \cdot (l_1 \rightsquigarrow \dots \rightsquigarrow l_n)$.

For reasons, which become apparent in the next subsection, it is convenient to denote $\alpha \cdot (l_1 \rightsquigarrow \dots \rightsquigarrow l_n)$ as $l_1 \rightsquigarrow \dots \rightsquigarrow l_n \rightsquigarrow \alpha$.

3.2. Finite prefix trees with numerical leaves

One can think of $l_1 \rightsquigarrow \dots \rightsquigarrow l_n$ as a path in a prefix tree (*trie*), with intermediate nodes being labeled by letters from L. So when one considers $\alpha \in \mathbb{R}$, one can express the presence of term $\alpha \cdot (l_1 \rightsquigarrow \dots \rightsquigarrow l_n)$ in our linear combination as presence of path with the intermediate nodes labeled by $l_1, \dots, l_n$ and the leaf labeled by α. We denote this path as $l_1 \rightsquigarrow \dots \rightsquigarrow l_n \rightsquigarrow \alpha$. Because we have finite linear combinations (terms and paths corresponding to $\alpha = 0$ tend to be omitted), we are talking about *finite prefix trees with numerical leaves.*

Example. The linear combination $3.5 \cdot (\varepsilon) + 2 \cdot (\texttt{:foo}) + 7 \cdot (\texttt{:foo :bar}) - 4 \cdot (\texttt{:baz :foo :bar})$, i.e., $(\rightsquigarrow 3.5) + (\texttt{:foo} \rightsquigarrow 2) + (\texttt{:foo} \rightsquigarrow \texttt{:bar} \rightsquigarrow 7) + (\texttt{:baz} \rightsquigarrow \texttt{:foo} \rightsquigarrow \texttt{:bar} \rightsquigarrow -4)$.

[3] the space of functions $f : L^* \to \mathbb{R}$ such that $f(w) \neq 0$ for no more than finite number of $w \in L^*$; the operations are pointwise: $(f + g)(w) = f(w) + g(w)$ and $(\alpha f)(w) = \alpha f(w)$

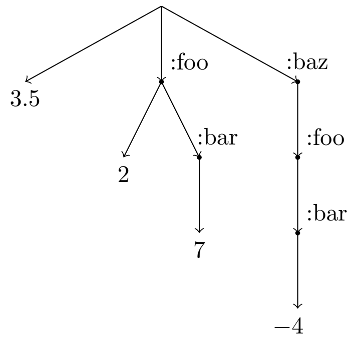


Figure 3: The prefix tree for this example

The empty string ε with non-zero coefficient β, written as $\beta \cdot (\varepsilon)$ or simply $\rightsquigarrow \beta$, corresponds to the leaf with non-zero β attached directly to the root of the tree.

The prefix tree (trie) for this example is shown in Figure 3. In this particular example, we label intermediate nodes with Clojure keywords, which start with ":" character. This example reminds us about the earlier remark that one can often think about letters from alphabet L as words from some language defined over some other, more conventional alphabet, and that this allows us to think about meaningful languages of labels.

3.3. Sparse "tensors of mixed rank"

Yet another "depth-first" way of looking at this situation is to consider $l_1 \rightsquigarrow \ldots \rightsquigarrow l_n \rightsquigarrow \alpha$ to be an element of sparse multidimensional array with n dimensions. E.g., $l_1 \rightsquigarrow l_2 \rightsquigarrow \alpha$ is an element of a sparse matrix with its row labeled by l_1 and its column labeled by l_2 and α being the value of the element.

The non-zero leaf attached directly to the root of the tree, $\beta \cdot (\varepsilon) = \rightsquigarrow \beta$, is considered to be a scalar with value β.

Each string of length one with non-zero coefficient γ, that is $l \rightsquigarrow \gamma$ (the leaf with non-zero γ attached to the end of the path of length one with the intermediate node in the path labeled by l), is considered to be a coordinate of a sparse array, where the coordinate is labeled by l and has value γ.

Each string (path) of length three with non-zero coefficient γ', written as $l_1 \rightsquigarrow l_2 \rightsquigarrow l_3 \rightsquigarrow \gamma'$, is considered to be an element of sparse three-dimensional array, etc.

The standard convention in machine learning is to call multidimensional arrays with n dimensions "tensors of rank n". Because our linear combinations generally include sequences from L^* of different lengths, we have to talk about *sparse "tensors of mixed rank"*. For example, the vector $(\rightsquigarrow 3.5) + ($`:foo` $\rightsquigarrow 2) + ($`:foo` $\rightsquigarrow$ `:bar` $\rightsquigarrow 7) + ($`:baz` $\rightsquigarrow$ `:foo` $\rightsquigarrow$ `:bar` $\rightsquigarrow -4)$ from the previous subsection is the sum of scalar 3.5, sparse array with one non-zero element `d1[:foo] = 2`, sparse matrix with one non-zero element `d2[:foo, :bar] = 7`, and sparse three-dimensional array with one non-zero element `d3[:baz, :foo, :bar] = -4`, so it is a typical "tensor of mixed rank".

Therefore all usual vectors, matrices, and tensors of any dimension can be represented in V. This is convenient from the viewpoint of machine learning as multidimensional tensors often occur in the machine learning practice.

The space V is a direct sum of the one-dimensional space of scalars and spaces of n-dimensional arrays: $V = V_0 \oplus V_1 \oplus V_2 \dots$ Since L is countably infinite, $V_1, V_2, \dots$ are infinite-dimensional vector spaces. If L is finite and consists of $\text{Card}(L)$ elements, then the dimension of V_i as a vector space is $\text{Card}^i(L)$.

This is also a good point to transition to "breadth-first" representations. A sparse matrix can also be viewed as a map from elements of L labeling its non-zero rows to the sparse vectors representing those rows (these sparse vectors are maps from elements of L labeling columns of the matrix to the values of the actual non-zero matrix elements; zero elements are omitted, as usual).

3.4. Recurrent maps

To obtain a "breadth-first" representation for a general element $v \in V$, we first note that there is a possibility that the $\alpha \cdot (\varepsilon) = \rightsquigarrow \alpha$ belongs to v with non-zero coefficient α, in which case the corresponding coordinate of v is a non-zero "tensor of rank 0", i.e., the scalar with value α.

Then for each letter $l_1 \in L$, such that some non-zero term $l_1 \rightsquigarrow l_2 \rightsquigarrow \dots \rightsquigarrow l_n \rightsquigarrow \beta$ belongs to v, we consider all terms from v which share the same first letter l_1, namely $l_1 \rightsquigarrow l_2 \rightsquigarrow \dots \rightsquigarrow l_n \rightsquigarrow \beta,\ l_1 \rightsquigarrow l'_2 \rightsquigarrow \dots \rightsquigarrow l'_m \rightsquigarrow \gamma, \dots$, and consider $v^{l_1} \in V$ consisting of those terms with the first letter removed, namely $l_2 \rightsquigarrow \dots \rightsquigarrow l_n \rightsquigarrow \beta,\ l'_2 \rightsquigarrow \dots \rightsquigarrow l'_m \rightsquigarrow \gamma, \dots$

We map each such letter l_1 to v^{l_1}, and we map each letter $l \in L$ for which v does not have a non-zero term starting from l to zero vector (zero element of V). The finite description of our map only needs to include the finite set of $\langle l_1, v^{l_1} \rangle$ pairs, and pairs $\langle l, 0 \rangle$ and $\langle l_1, 0 \rangle$ can all be omitted.

An element of v is then a pair consisting of a scalar and a map from L to V admitting a finite description. Either or both elements of this pair can be zero.

As a vector space, V satisfies the following equation:

$$V \cong \mathbb{R} \oplus (L \to V). \tag{3}$$

Here $L \to V$ is a space of such maps from L to V that only a finite number of elements of L map to non-zero elements of V. Let us call such maps *finitary*. This equation is an *isomorphism of vector spaces*. It reflects the fact that every element $v \in V$ can be represented as a pair of $\alpha \in \mathbb{R}$ and finitary map $l \mapsto v^l$, as was shown earlier in the present subsection, and vice versa every pair consisting of $\alpha \in \mathbb{R}$ and a finitary map $L \to V$ is obtained in this fashion.

We would like to be able to represent elements of V not by pairs of a number and a map, but simply by maps. To include numbers α into the map itself, we would need a separate label for them that does not appear in L. Thus we take a new key $n \notin L$ and $L' = L \cup \{n\}$, then represent $\langle \alpha, \{\langle l_1, v^{l_1} \rangle, \ldots, \langle l_n, v^{l_n} \rangle\} \rangle$ as $\{\langle n, \alpha \rangle, \langle l_1, v^{l_1} \rangle, \ldots, \langle l_n, v^{l_n} \rangle\}$.

Since $\mathbb{R}$ is embedded into V via representation of $\alpha \in \mathbb{R}$ as $\langle \alpha, 0 \rangle$, space V is isomorphic to a subspace of $L' \to V$. This isomorphism is why we call this space a space of *recurrent maps*: every element of V is represented as a finitary map from extended alphabet L' to the space V itself.

This representation of V via finitary maps to V is fundamental to our constructions in the present paper, because it translates directly to our Clojure implementation of core DMMs primitives (DMM 2016–2017) and because it allows us to introduce *variadic neurons*.

Implementation. We implement elements $v \in V$ as recurrent maps. Usually, programming languages provide *dictionaries* or *hash-maps* suitable for this purpose. In our Clojure implementation, elements $v \in V$ are represented by hash-maps, which map elements of L' to V.

Typically, L will be the set of all legal hash-map keys available in our language with the exception of a few keys reserved for other purposes. In particular, we reserve Clojure keyword `:number` to be mapped into the scalar component of a pair $\langle \alpha, \{\langle l_1, v^{l_1} \rangle, \ldots, \langle l_n, v^{l_n} \rangle\} \rangle$. So, in our case $L' = L \cup \{\texttt{:number}\}$.

Therefore, $\langle \alpha, \{\langle l_1, v^{l_1}\rangle, \ldots, \langle l_n, v^{l_n}\rangle\}\rangle$ is represented in Clojure by the hash-map `{:number` $\alpha, l_1\ v^{l_1}, \ldots, l_n\ v^{l_n}$`}`.

Here $v^{l_1}, \ldots, v^{l_n}$ are represented by similar hash-maps themselves resulting in *nested hash-maps*, and keys from L can have rather complex structure, if desired, taking advantage of great variety of hash-map keys allowed in Clojure.

When element $v \in V$ is simply a scalar (the pair $\langle \alpha, 0\rangle$), the implementation is allowed to simply use number α instead of the hash-map `{:number` α`}`.

Example from section 3.2. The sum ($\rightsquigarrow$3.5) + (`:foo` $\rightsquigarrow$ 2) + (`:foo` $\rightsquigarrow$ `:bar` $\rightsquigarrow$ 7) + (`:baz` $\rightsquigarrow$ `:foo` $\rightsquigarrow$ `:bar` $\rightsquigarrow$ -4) is represented as Clojure hash-map `{:number 3.5, :foo {:number 2, :bar 7}, :baz {:foo {:bar -4}}}`.

Variadic Neurons. We use the formalism of V-values to eliminate the need to keep track of the number of input and output arguments of the activation functions. We describe variadic neurons and DMMs based on variadic neurons in section 4.

To conclude the present section, V-values are essentially a dictionary-based version of S-expressions. Section 5.3 removes the restriction that all atoms must be numbers and allows to incorporate complex objects under reserved keywords.

4. Variadic neurons

The activation functions of variadic neurons transform a single stream of V-values into a single stream of V-values.

However, the labels at the first level of those V-values are dedicated to serve as the names of input and output arguments. Therefore, a neuron is a priori *variadic* and can potentially handle a countable collection of inputs and produce a countable collection of outputs (although our usual restrictions of keeping the active part of the network finite would in practice limit those collections to finite at any given moment of time).

Here is an example of an activation function for neuron with two arguments, x and y, outputting two results, difference $(x-y)$ and negative difference $(y-x)$. This and all subsequent code examples in this paper are written in Clojure.[4] Assume for the purpose of this example that `my-minus`

[4] https://clojure.org

function is available to compute these subtractions of one V-value from another V-value:[5]

```
(defn symmetric-minus [input]
  (let [x (get input :x {})
        y (get input :y {})]
    {:difference (my-minus x y)
     :negative-difference (my-minus y x)}))
```

The `input` is a hash-map, representing a V-value. The arguments are subtrees of `input` corresponding to `:x,:y` $\in L$. These subtrees are computed by the expressions `(get input :x {})` and `(get input :y {})`. If the subtree in question is not present, the empty hash-map `{}` is returned. The empty hash-map represents zero vector (zero element of V) in our implementation. The function outputs a V-value with two subtrees corresponding to `:difference,:negative-difference` $\in L$.

So, the arguments are combined into a single V-value, `input`, and the outputs are combined into a single return V-value.

The non-trivial aspect of this approach is that instead of mapping neuron outputs to neuron inputs, the network matrix **W** now maps subtrees at the first level of the neuron outputs to subtrees at the first level of the neuron inputs. We shall see in the present section that the network matrix **W** naturally acquires a structure of multidimensional tensor under this approach.

4.1. Space *U*

Because we are going to use the keys at the first level of V-values as names of inputs and outputs, we do not want any top-level leaves, that is we do not want non-zero scalars ("tensors of rank 0") in our V-values. Hence we will be using space $U = L \to V$, namely we'll use values $u \in U$ as *inputs and outputs of the neuron activation function* (which will always have arity one), and those values u will contain *actual inputs and outputs of the neuron* at their first level.

Observe that $V \cong \mathbb{R} \oplus (L \to V) = \mathbb{R} \oplus U$, and that U therefore satisfies the equation $U \cong L \to (\mathbb{R} \oplus U)$.

[5] Generally, one would want to define `my-minus` in terms of DMM core primitives for V-values implemented in DMM (2016–2017): `(defn my-minus [x y] (rec-map-sum x (rec-map-mult -1 y)))`.

The activation functions of the neurons map U to U, transforming single streams of elements of U. The labels at the first level of the elements of U serve as names of inputs and outputs.

4.2. Multidimensional structure on *W*

The network matrix $\mathbf{W}$ must provide a linear map from the concatenation of the first levels of elements of U which are the outputs of all neurons, to the concatenation of the first level of elements of U which are the inputs of all neurons. In our example above, a neuron with `symmetric-minus` activation function would have two V-values at its input, one labeled by `:x` and one labeled by `:y`, assembled into one input map. Such a neuron would also have two V-values at its output, one labeled by `:difference` and one labeled by `:negative-difference`, assembled into one output map.

However, we consider an infinite collection of V-values on input of each neuron, with V-values labeled by all elements of L, and an infinite collection of V-values on output of each neuron, with V-values labeled by all elements of L, since nothing restricts activation functions from using any of those labels.

Now we want to take infinite collections of V-values on *output* of each neuron for all neurons and join those infinite collections together into a single infinite collection of V-values, and we shall apply matrix $\mathbf{W}$ to this unified infinite collection (imposing the usual condition about only a finite number of relevant elements or vectors being non-zero at any given moment of time).

We also want to take infinite collections of V-values on *input* of each neuron for all neurons and join those infinite collections together into a single infinite collection of V-values, and matrix $\mathbf{W}$ will produce this unified infinite collection each time it is applied to the collection described in the previous paragraph.

Below we follow closely the material from section 3.2 of Bukatin & Anthony (2017).

Consider *one input* situated on the first level of the element of U serving as the argument of the activation function for one neuron, and the row of the network matrix $\mathbf{W}$ responsible for computing *that input* from the concatenation of the first levels of elements of U which are the outputs of all neurons.

The natural structure of indices of this row is not flat, but hierarchical. At the very least, there are two levels of indices: neurons and their outputs.

We currently use three levels of hierarchy in our implementation: neuron types (which are Clojure *vars* referring to implementations of activation functions $U \to U$), neuron names, and names of the outputs. Hence, matrix rows are three-dimensional sparse arrays (sparse "tensors of rank 3") in our current implementation.

The natural structure of indices of the array of rows is also not flat, but hierarchical. At the very least, there are two levels of indices: neurons and their inputs. We currently use three levels of hierarchy in our implementation: neuron types, neuron names, and names of the inputs.

Hence, the network matrix $\mathbf{W}$ is a six-dimensional sparse array (sparse "tensor of rank 6") in our current implementation.

4.3. DMMs based on V-values and variadic neurons

The DMM is a "two-stroke engine" similar to that of section 2, and consists of a linear "down movement" followed by an "up movement" performed by the activation functions of neurons. This "two-stroke cycle" is repeated indefinitely (Figure 4).

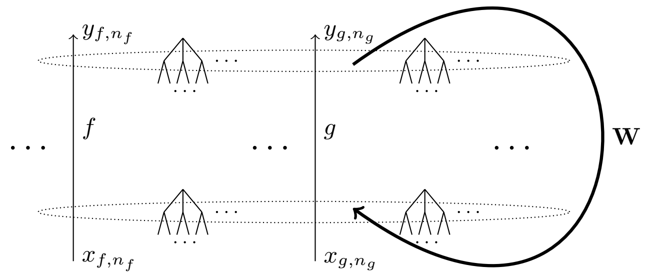


Figure 4: "Two-stroke engine" for a DMM based on variadic neurons. Two neurons, n_f and n_g, are explicitly pictured. Their inputs and outputs, $x_{f,n_f}, x_{g,n_g}, y_{f,n_f}, y_{g,n_g}$, are depicted as trees belonging to U. $\mathbf{W}$ is a linear map from the concatenation of the first levels of all y_{h,n_h} trees to the concatenation of the first levels of all x_{h,n_h} trees (Equation 4).

We allow names for neurons and their inputs and outputs to be any elements of L. The address space is such that the network is countably-sized, but since the network matrix $\mathbf{W}$ has only a finite number of non-zero elements at any given time, and elements of U have only a finite number of non-zero coordinates at any given time, we are always working with finite representations.

The network matrix **W** (sparse "tensor of rank 6") depends on t, and its element $w^t_{f,n_f,i;\,g,n_g,o}$ is non-zero, if the output o of neuron n_g with the built-in activation function g is connected to the input i of neuron n_f with the built-in activation function f at the moment of time t, with number $w^t_{f,n_f,i;\,g,n_g,o}$ being the non-zero weight of this connection.

On the "down movement", the network matrix $(w^t_{f,n_f,i;\,g,n_g,o})$ is applied to an element of U which contains all outputs of all neurons. The result is an element of U which contains all inputs of all neurons to be used during the next "up movement". Each of those inputs is computed using the following formula:

$$x^{t+1}_{f,n_f,i} = \sum_{g \in F} \sum_{n_g \in L} \sum_{o \in L} w^t_{f,n_f,i;\,g,n_g,o} * y^t_{g,n_g,o}. \tag{4}$$

Indices f and g belong to the set of neurons types F, which is simply the set of transformations of U. Potentially, one can implement countable number of such transformations in a given programming language, but at any given time only finite number of them are defined and used. Indices n_f and n_g are the names of input and output neurons, and indices i and o are the names of the respective input and output arguments of those neurons.

In the formula above, $w^t_{f,n_f,i;\,g,n_g,o}$ is a number (the connection weight), and $x^{t+1}_{f,n_f,i}$ and $y^t_{g,n_g,o}$ are elements of V (not necessarily of U, since the presence of scalars is allowed at this level). The product $w^t_{f,n_f,i;\,g,n_g,o} * y^t_{g,n_g,o}$ multiplies vector $y^t_{g,n_g,o}$ by real number $w^t_{f,n_f,i;\,g,n_g,o}$.

This operation is performed for all $f \in F$, all $n_f \in L$, all input names $i \in L$, such that the corresponding three-dimensional row of **W** has some non-zero elements at time t.

The result is finitely sized map $\{f \mapsto \{n_f \mapsto x^{t+1}_{f,n_f}\}\}$, where each x^{t+1}_{f,n_f} is a finitely sized map from the names of neuron inputs to the values of those inputs, $\{i \mapsto x^{t+1}_{f,n_f,i}\}$.

On the "up movement", each f is simply applied to the elements of U representing the single inputs of the activation function f. This application is performed for all neurons $\langle f, n_f \rangle$ which are present in the finitely sized map described in the previous paragraph:[6]

[6] The neuron is fully determined by a pair $\langle f, n_f \rangle$. For each $f \in F$, there can be many active neurons $\langle f, n_f \rangle$ with different n_f. For each $n_f \in L$, there can be many active neurons $\langle f, n_f \rangle$ with different f.

$$y^{t+1}_{f,n_f} = f(x^{t+1}_{f,n_f}). \tag{5}$$

An example of an activation function of a variadic neuron was given in the beginning of the present section. For more examples of this kind see section 6.1 and section 6.3.

We gave the construction for the space V, but similar constructions also work for more general variants described in section 5.3.

5. Linear streams

Informally speaking, we say that a space of streams is a *space of linear streams*, if a meaningful notion of linear combination of several streams with numerical coefficients is well-defined.

We would like to formalize this notion, while keeping the following examples in mind:

- The space of sequences of numbers;
- For a vector space V, the space of sequences of its elements, $(v_1, v_2, \ldots)$;
- For a measurable space X, spaces of samples and signed samples drawn from X.

We consider linear streams in somewhat limited generality here. First of all, we consider discrete sequential time, while other models of time, e.g., continuous, are also potentially of interest.[7] We consider linear streams over real numbers, while other systems of coefficients (especially, complex numbers) can also be quite fruitful. Finally, we ground each space of linear streams in a vector space, while one could consider a more abstract approach to the notion of linear combination, where one works solely with streams of abstract representations without grounding them in a vector space.

To define a particular *kind* of linear streams k in a more formal manner, we specify background vector space V_k and streams of *approximate representations* of the underlying vectors from V_k. The approximate representations provide some information about the underlying vectors. Moreover, for every kind of linear streams k, we specify a procedure computing an approximate representation of a linear combination $\alpha_1 v_{1,k} + \ldots + \alpha_n v_{n,k}$ from

[7] We understand streams as functions from time to a set of objects, so that an object corresponds to any given moment of time. In our paper, time tends to have a starting point, to be discrete, and to continue indefinitely, so we usually model time by non-negative integers starting from 0 or 1.

approximate representations of vectors $v_{1,k}, \ldots, v_{n,k}$. We say informally that the approximate representations in question belong to a vector-like space and we call them *vector-like elements*.

5.1. Streams of samples and signed samples

First, let us consider streams of samples from sequences of probability distributions over an arbitrary measurable space X and their linear combinations with positive coefficients.

A sequence of probability distributions, $(\mu_1, \mu_2, \ldots)$, can be represented by a sequence of elements of X sampled from those distributions, $(x_1, x_2, \ldots)$. Note that X is not required to be a vector space. Consider $0 < \alpha < 1$ and sequences of probability distributions $(\mu_1, \mu_2, \ldots)$ and $(\nu_1, \nu_2, \ldots)$ represented by streams of samples $(x_1, x_2, \ldots)$ and $(y_1, y_2, \ldots)$. Produce a stream of samples representing the sequence of probability distributions $(\alpha * \mu_1 + (1-\alpha) * \nu_1, \alpha * \mu_2 + (1-\alpha) * \nu_2, \ldots)$ as follows. Sample a random number uniformly from [0,1], and if this number is smaller than α, pick x_1 as the first element of our stream, otherwise pick y_1. Repeat this procedure for x_2 and y_2, and so on. Let us call this procedure a *stochastic linear combination* of the streams of samples.

To include this example into our framework, we need a background vector space, and we need stochastic linear combinations with positive and negative coefficients to be well-defined.

Let us observe that probability distributions over some measurable space X belong to the vector space of finite signed measures[8] over X. Let us consider *signed samples*, that is, samples marked as being positive or negative.

Streams of signed samples are mentioned in section 1.2 of Bukatin & Matthews (2015c). Here we follow a more detailed treatment of them given in Appendix A.1 of Bukatin & Anthony (2017). The underlying vector space is the space of all finite signed measures over arbitrary measurable space X, and samples are pairs $\langle x, s \rangle$, with $x \in X$ and flag s taking values -1 and 1.

One considers streams of finite signed measures over X, $\mu_1, \ldots, \mu_n$, and streams of corresponding samples, $\langle x_1, s_1 \rangle, \ldots, \langle x_n, s_n \rangle$.

The procedure of computing a sample representing a signed measure $\alpha_1 * \mu_1 + \ldots + \alpha_n * \mu_n$ is as follows. We pick index i with probability $\mid \alpha_i \mid / \sum_j \mid \alpha_j \mid$ using the absolute values of coefficients α, and we pick

[8] measures which can take any finite real values, including negative values

the sample $\langle x_i, \text{sign}(\alpha_i) * s_i \rangle$ (reversing the flag if the selected value α_i is negative) to represent the linear combination $\alpha_1 * \mu_1 + \ldots + \alpha_n * \mu_n$.

For further discussion of expressive power of signed measures and signed samples see section 1.2 of Bukatin & Matthews (2015c). Issues related to missing samples and zero measures are mentioned in Appendix A.2 of Bukatin & Anthony (2017). A generalization to complex-valued measures and linear combinations with complex coefficients is considered in the design notes for DMM project (see DMM 2016–2017).[9]

5.2. Embedding

The ability to represent characters, words, and other objects of discrete nature as vectors is one of the cornerstones of success of modern neural networks.

The embedding of characters into a vector space generated by the alphabet ("one-hot encoding") is a basis for rather spectacular results obtained by modern forms of recurrent neural networks, such as LSTMs (Karpathy 2015). The ability to learn an optimal embedding of words into vectors is an integral part of a number of applications of recurrent neural nets to linguistics (Mikolov et al. 2013).

When it comes to representing compound structures in vector spaces, there is obviously a lot of freedom and variety. Throughout this paper, we work with embeddings of classes of dataflow matrix machines (considering each time a class of DMMs over some signature of neuron types) into corresponding vector spaces. One can argue that a class of dataflow matrix machines forms a sufficiently rich space of objects, and that since one finds a meaningful natural embedding of such a space into a vector space, one should expect to be able to find meaningful natural embeddings for a large variety of spaces of objects.

As the previous subsection indicates, the notion of embedding into linear streams is more general than the notion of embedding into vector spaces. As long as one is willing to consider a stream of objects of arbitrary nature as drawn from some sequence of probability distributions over those objects, this stream belongs to a space of linear streams of samples equipped with the stochastic version of linear combination.

Hence one is able to obtain an embedding of a stream of objects into a space of linear streams without embedding individual objects into a vector space.

[9] https://tinyurl.com/y8bfwmre

5.3. A family of spaces of V-values

The space V is very expressive, but sometimes it is not enough. If one wants to accommodate vectors from some other vector space V', a convenient way to do so is to allow elements of V' in the leaves of the prefix trees. Usually one still wants to be able to have just numbers inside the leaves as well, so one uses $\mathbb{V} = \mathbb{R} \oplus V'$ for the space of leaves.

This means that elements of $\mathbb{V}$ are now used as coefficients α instead of real numbers in $\alpha \cdot (l_1 \dots l_n) = l_1 \rightsquigarrow \dots \rightsquigarrow l_n \rightsquigarrow \alpha$. Considering a basis in $\mathbb{V}$, one can see that this whole construction corresponds to tensor product $V \otimes \mathbb{V}$.

Elements of $\mathbb{V}$ are also used as elements of sparse "tensors of mixed rank". The equation $V \cong \mathbb{R} \oplus (L \to V)$ becomes $V \cong \mathbb{V} \oplus (L \to V)$.

Implementation-wise, if, for example, $\mathbb{V} = \mathbb{R} \oplus V' \oplus V'' \oplus V'''$, then in addition to the reserved key `:number`, one would reserve additional keys for each of the additional components V', V'', V''' to incorporate the non-zero instances of those components into leaves (and into hash-maps).

For example, Appendix A.3 of Bukatin & Anthony (2017) shows how to accommodate signed samples discussed in section 5.1 above within this framework. One considers $\mathbb{V} = \mathbb{R} \oplus M$, where M is the space of finite signed measures over X, and one uses the reserved keyword `:sample` to incorporate signed samples into the leaves as necessary.

So, in this fashion a space $V \cong \mathbb{V} \oplus (L \to V)$ will still be represented by nested hash-maps, and we are still going to call those nested hash-maps *V-values.*

6. Programming

Here we discuss some of the programming patterns in dataflow matrix machines. Further programming tools come from the presence of self-referential mechanism (section 7) and from modularization facilities (section 9).

6.1. Linear and multiplicative constructions

Linear and multiplicative constructions in dataflow matrix machines are well-covered in Bukatin et al. (2016d). The most fundamental of them is a neuron with *identity activation function.* Consider some argument name, e.g., `:accum`. If we connect output `:accum` of such a neuron to its input `:accum` with weight 1, this neuron becomes an *accumulator*. It adds together and accumulates in its `:accum` arguments the contributions to its input `:accum` made during each "two-stroke cycle" by all other outputs in the network connected to the input `:accum` of our neuron by non-zero weights.

Among multiplicative constructions, the most fundamental one is multiplication of an otherwise computed neuron output by the value of one of its scalar inputs. This is essentially a fuzzy conditional, which can selectively turn parts of the network on and off in real time via multiplication by zero, attenuate or amplify the signal, reverse the signal via multiplication by -1, redirect flow of signals in the network, etc. For further details see Bukatin et al. (2016d).

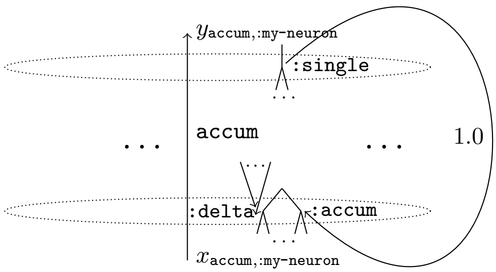


Figure 5: Connectivity of a neuron `[accum :my-neuron]` with activation function `accum`, neuron name `:my-neuron`, and input arguments `:accum` and `:delta`, when this neuron is used as an accumulator

These facilities are quite powerful even for scalar flows of reals, and even more so for vector flows. The lack of these facilities hinders the approaches which insist on only having non-linear activation functions, or on only having neurons with a single input.

Sometimes, it is convenient to take advantage of having multiple inputs and use a neuron with $+$ activation function, $y = x + \Delta x$, as an accumulator, connecting y to x with weight 1 and accepting contributions from other outputs in the network on input Δx (Bukatin et al. 2016c). For details of this connectivity pattern in the current DMM architecture with variadic neurons see Figure 5 .

The code for the activation function `accum` of this accumulator looks as follows (DMM 2016–2017):

```
(defn accum [input]
    {:single (rec-map-sum (input :accum) (input :delta))})
```

The function `accum` takes a V-value `input` as argument, then it adds together the values of the two neuron inputs,[10] `(input :accum)` and `(input :delta)`, obtaining the desired result, and then returns `{:single result}` as the output. It can add together arbitrary elements of V.

In this example, the `:accum` name for a neuron input stands for x, the `:delta` name for a neuron input stands for Δx, and the `:single` is the name of the only output this particular neuron has.

If we want a particular neuron `:my-neuron` with built-in activation function `accum` to work as an accumulator, then the element of the network matrix connecting the output `[accum :my-neuron :single]` together with the input `[accum :my-neuron :accum]` should be 1 (see Figure 5). The corresponding matrix element $w_{\texttt{accum,:my-neuron,:accum; accum,:my-neuron,:single}}$ is expressed as

```
{v-accum {:my-neuron {:accum
  {v-accum {:my-neuron {:single 1}}}}}}
```

in our current Clojure implementation, where `v-accum` is defined as

```
(def v-accum (var accum))
```

because the value of `accum` itself used as a hash-key is not stable from recompilation to recompilation, whereas `(var accum)` is stable.

The contributions from other neurons are accepted on the input `:delta`.

6.2. Sparse vectors

The ability to handle sparse vectors in a straightforward manner is not available in scalar-based neural networks. One has to allocate a neuron for every coordinate, and there is no straightforward way to avoid processing those coordinates which happen to be zero at the moment.

In DMMs one can handle compact sparse vectors of very high, or even infinite dimensions, and the network size does not depend on those dimensions. An example of a compact DMM built around an accumulator

[10] `rec-map-sum` is one of DMM core primitives for V-values implemented in DMM (2016–2017) which performs addition of V-values

of sparse vectors in the vector space generated by a given alphabet in order to keep track of the number of occurrences of each character in a given string is studied in detail in Bukatin et al. (2016d). The savings can be drastic, as, for example, Unicode alphabet exceeds 100,000 characters, and with sparse representation only the actually occurring characters are stored and processed.

The same technique can be applied to keeping track of the number of occurrences of each word in the text, where the underlying vector space generated by all possible words is infinitely-dimensional.

6.3. Data structures

An earlier paper (Bukatin et al. 2016d) focuses on allocating data structures in the body of the network itself, an approach encouraged by the network's potentially infinite size and powerful self-modification facilities.

However, with the ability to process complicated vectors such as V-values, it is natural to encode and process data structures on that level. The use of structure sharing immutable data structures, the default in Clojure, should make passing complicated structures through "up movements" and "down movements" reasonably efficient, as seen in our own preliminary explorations.

For example, a list can be encoded by using `:this` and `:rest` keys on the same level of a V-value, and having neurons with `first`, `rest`, and `cons` activation functions. The only decision one needs to make is whether to consider all lists as having infinite sequences of zeros at the tail, or whether to incorporate an explicit list terminator (e.g., a keyword `:end` mapped to `1`) in the formalism.

For example, the following activation function for the neuron used to accumulate a list of "interesting" events is similar to the function used in our example of a DMM accumulating a list of mouse clicks:[11]

```
(defn dmm-cons [accum-style-input]
  (let [old-self (get accum-style-input :self {})
        new-signal (get accum-style-input :signal {})]
    (if (interesting? new-signal)
      {:self {:this new-signal :rest old-self}}
      {:self old-self})))
```

In order to maintain the accumulator metaphor, the `:self` output and the `:self` input of the corresponding neuron are connected with weight 1.

[11] https://tinyurl.com/yxgagfjl

Any linked structures which can be encoded inside the network matrix, can be encoded inside a similar matrix not used as the network matrix (the only difference is that data structures encoded within the network matrix tend to be "active", as they are built over actively working neurons; this difference can potentially be profound).

6.4. From programming via composition of transformers of streams of V-values to Dataflow Matrix Machines

There is a rather long history of programming via composition of transformers of linear streams. In each of those cases, linear streams can be represented as sufficiently general streams of V-values.

Perhaps the most well known example is the discipline of audio synthesis via composition of *unit generators*, which are transformers of streams of audio samples (streams of numbers, if one considers a single monophonic channel). That discipline was created by Max Mathews in 1957 at Bell Labs (Mathews 1963). It is typical for a modern audio synthesis system to be crafted along those lines, even though the syntax can differ greatly. One of the classical textbooks in that discipline is Farnell (2010). We found the tutorial (Merz 2011) on Beads, a realtime audio and music library for Java and Processing, to be a convenient introductory text.

In this cycle of studies we explored a number of different examples of programming via composition of linear streams, starting from dataflow programming of animations via composition of transformers of image streams (Bukatin & Matthews 2015a). During that series of experiments we discovered that if one does not want to impose the condition of the dependency graph being acyclic, then one needs to maintain two elements of each stream at any given time, the "current" element, used by the transformers depending on that stream in their computations, and the "next" element, the element which is being computed. Then, after transformers computed their respective "next" elements, there is a *shift* operation which makes all the "next" elements current.

This cycle "transform-shift" is a version of the two-stroke cycle used in this paper, and the shift operation is what the "down movement" of a DMM would look like, if all non-zero matrix rows would only have a single non-zero element in each of them, and that element would be 1. In fact, any program built as a composition of transformers of linear streams can be converted into an equivalent DMM by inserting a linear transformation described by a matrix row with one non-zero element with weight 1 at each connection.

A number of examples of DMMs we explored in this cycle of studies, such as the character-processing DMM described in section 3 of Bukatin et al. (2016d) or the DMM accumulating a list of mouse clicks mentioned in section 6.3 of the present paper, come from programs built as compositions of transformers of linear streams converted to DMMs by inserting weight 1 connectors.

One aspect which used to be somewhat limited within the discipline of programming via composition of transformers of linear streams was higher-order programming, i.e., transforming the transformers. In particular, the higher-order constructions themselves were not expressed as transformations of linear streams.

Dataflow matrix machines allow to continuously transform any composition of stream transformers into any other composition of stream transformers, and we present one way to do so *within the discipline of transforming linear streams* and in a self-referential manner in the next section.

7. Self-referential mechanism

The ability to handle arbitrary linear streams implies the ability to handle streams of vectors shaped like network connectivity matrices (be those flat two-dimensional matrices, or sparse multidimensional tensors described in section 4). This enables a rather straightforward mechanism to access and modify the network matrix $\mathbf{W}$. We designate a neuron `Self` emitting a stream of such matrices, and use the most recent value from that stream as $\mathbf{W}$ for the purpose of the next "down movement" step.

We currently prefer to use an accumulator with + activation function $y = x + \Delta x$ as `Self` following Bukatin et al. (2016c) (see section 6.1). `Self` takes additive updates from other neurons in the network on its Δx input, and other neurons can take the stream of the current values of $\mathbf{W}$ from the output of `Self` making them aware of the current state of the network connectivity.

Network self-modification based on the streams of network matrices was first introduced in Bukatin et al. (2016b), and the principle of "self-referential completeness of the DMM signature relative to the language available to describe and edit the DMMs" was formulated there. That principle states that it is desirable to have a sufficient variety of higher-order neurons to perform updates of the network matrix, so that any modifications of a DMM (understood as a network or as a program) can be made by triggering an appropriate higher-order neuron.

Paper Bukatin et al. (2016d) explored ways for the network to modify itself by making deep copies of its own subgraphs. The possibility of using self-referential matrix transformations as a new foundation for programming with linear streams, somewhat similarly to lambda-calculus being a foundation for symbolic programming, was studied in Bukatin et al. (2016c).

Also, it was demonstrated in Bukatin et al. (2016c) that this self-referential mechanism together with a few constant update matrices gives rise to a wave pattern dynamically propagating in the network matrix; this result was also verified in computer experiments (see Appendix B of Bukatin & Anthony (2017) for a more polished presentation).

However, all these studies were so far merely scratching the surface of what is possible with this mechanism. In principle, it should allow the network to maintain an evolving population of its own subnetworks, to maintain an evolving population of network update methods, to train network update methods as a linear combination of available network update primitives, etc.

We hope that some of this potential will be explored in the future.

8. Network topology

The network topology, such as layers, is defined by the pattern of sparsity of $\mathbf{W}$: some of the connectivity weights are kept at zero, and some are allowed to deviate from zero, and the network topology comes from that.

However, in the model of synchronous time we follow in this paper, all layers still work simultaneously. If it is desirable for layers to work strictly in turn, one can use multiplicative mechanisms described in section 6.1 to orchestrate the computations by turning off the layers at the appropriate moments of time using zero *multiplicative masks*, and then by further optimizing implementation to save processing time in such situations.

Not all weights which are non-zero need to be variable weights. It is often the case that some of the non-zero weights are set to 1, and then it is the user's choice which of those should be allowed to vary. The particularly frequent are the cases when the weight 1 in question is the only non-zero element of its matrix row. The examples of cases where weights set to 1 occur naturally include

– accumulators (sections 6.1, 6.3);
– cases when a program expressed as composition of V-value transformers is translated into a DMM (section 6.4);
– special neural network topology.[12]

9. Subnetworks and modularization

The modern trend in artificial neural nets is to build the networks not from a large number of single neurons, but from a relatively small number of modules such as layers, etc.

In this sense, it is convenient that neurons in DMMs are powerful enough to express the "up movement" action of whole subnetworks. This allows to build DMMs from a relatively small number of powerful neurons, if so desired.

In the old style DMMs (Bukatin et al. 2016a;b;c;d), the neurons had fixed arity and were powerful enough to express the "up movement" action of the subnetworks of fixed size. However, the networks themselves became variadic networks of unbounded size, so the gap between single neurons and general subnetworks remained.

With variadic neurons this particular gap is eliminated.

In 2016, Andrey Radul formulated a principle stating that there is no reason to distinguish between a neuron and a subnetwork, and that it is a desirable property of a model to not have a difference between a generalized neuron and a subnetwork.

The formalism of V-values and variadic neurons allows DMMs to *fulfill this principle in the following limited sense*: single neurons are powerful enough to express one "up movement" action of any subnetwork as one "up movement" action of an appropriately crafted single neuron.

[12] For example, let us express LSTMs and Gated Recurrent Unit networks as networks built from sigmoid neurons, linear neurons, and neurons performing multiplication for gating following Appendix C of Bukatin et al. (2016c), and let us assume that we are writing this in terms of neurons processing scalar streams (streams of numbers). Then each of the two inputs for each neuron performing multiplication of scalar streams is connected with weight 1 from a single output of an appropriate neuron.

10. Learning

There are various indications that dataflow matrix machines have strong potential for future machine learning applications.

DMMs contain well-known classes of neural networks with good machine learning properties, such as LSTM and Gated Recurrent Unit networks (Appendix C of Bukatin et al. 2016c).

At the same time, they allow to naturally express a number of various algorithms within a formalism which allows arbitrarily small modifications of programs, where one can transform programs continuously by continuously transforming the matrices defining those programs.

The presence of well-developed self-referential facilities means that network training methods can be made part of the network itself, making this a natural setting for a variety of "learning to learn" scenarios.

DMM architecture is conductive to experiments with "fast weights" (e.g., Ba et al. 2016).

Recently, we have been seeing very interesting suggestions that synthesis of small functional programs and synthesis of neural network topology from a small number of modules might be closely related to each other (Olah 2015; Nejati 2016). The ability of single DMM neurons to represent neural network modules suggests that DMMs might provide the right degree of generality to look at these classes of problems of network and program synthesis.

We are seeing evidence that syntactic shape of programs and their functionality provide sufficient information about each other for that to be useful during program synthesis by machine learning methods (e.g., Murali et al. 2017). If a corpus of human-readable programs manually written in the DMM architecture emerges eventually, this should be quite helpful for solving the problem of synthesis of human-readable programs performing non-trivial tasks.

Given that DMMs form a very rich class of computational models, it makes sense to search for its various subclasses for which more specialized methods of machine learning might be applicable.

11. Historical remarks and related work

There are two ways one can arrive at the dataflow matrix machines. One can focus on programming with linear streams and then notice that by interleaving linear and non-linear transformations of those streams, it is possible to obtain parametrization of large classes of programs by matrices of numbers.

Another way is to focus on recurrent neural networks as a programming platform and to try to generalize them as much as possible while retaining their key characteristic, which is parametrization by connectivity matrices of network weights.

In this section of the present paper we discuss some of the related work under both of these approaches.

11.1. Recurrent neural networks as programs

It was recognized as early as 1940s, that if one provides a neural network with a suitable model of unbounded memory one obtains a Turing-universal formalism of computations (McCulloch & Pitts 1943). Research studies formally establishing Turing-universality for neural networks processing streams of reals include Pollack (1987); Siegelmann & Sontag (1995).

More recent studies include such well-known approaches as Graves et al. (2014); Weston et al. (2014), which are currently under active development.

Yet, as these approaches are gradually becoming more successful at learning neural approximations to known algorithms, they do not seem to progress towards human-readable programs. In fact, it seems that while the expressive power of scalar-based neural networks is sufficient to create Turing-complete esoteric programming languages, they are not expressive enough to become a pragmatic programming platform. The restriction to scalar flows seems to either necessitate awkward encoding of complex data within reals (as in Siegelmann & Sontag 1995, where binary expansions of real numbers are used as tapes of Turing machines), or to force people to create networks depending in their size on data dimensionality and with any modularization and memory capabilities being external to the network formalism, rather than being native to the networks in question.

The awkward encodings of complex data within reals hinder the ability to use self-modification schemas for scalar-based neural networks. E.g., a remarkable early paper (Schmidhuber 1993) has to use such encodings for addresses in the network matrix, and such encodings lead to very high sensitivities to small changes of numbers involved, while the essence of correct neural-based computational schemas is their robustness in the presence of noise.

Even such natural constructions within the scalar flow formalism as neurons with linear activation functions, such as identity, and neurons performing multiplication of two arguments, each expressing a different linear combination of neuron outputs, encounter resistance in the field.

The power of linear and multiplicative neurons was well understood at least as early as 1987 (Pollack 1987). Yet, when the LSTMs were invented in 1997 (Hochreiter & Schmidhuber 1997), the memory and gating mechanisms were understood as mechanisms external to neural networks, rather then the mechanisms based straightforwardly on neurons with linear activation functions for memory and neurons performing multiplication for gating, which provide a natural way to express memory and gating mechanisms in neural nets (see Appendix C of Bukatin et al. 2016c).

Recently, the power of having linear activation functions, in particular identity, in the mix together with other activation functions is finally getting some of the recognition it deserves (the paper, He et al. 2016, is now a well-cited paper). However, the explicit activation functions of arity two, such as multiplication, are still quite exotic and often difficult to explicitly incorporate into existing software frameworks for neural networks.

We think that dataflow matrix machines as presented here, with their vector flows and multiple arities for activation functions, provide the natural degree of generality for neural networks.

11.2. Programming with linear streams

Continuous computations (which tend to be computations with linear streams) have a long history, starting with electronic analog computers. The programs, however, were quite discrete: a pair of single-contact sockets was either connected with a patch cord, or it was not connected with a patch cord.

More modern dataflow architectures focusing on work with linear streams representing continuous data include, for example, LabVIEW (Johnston et al. 2004) and Pure Data (e.g., Farnell 2010). The programs themselves are still quite discrete.

To incorporate higher-order programming methods within the paradigm of programming with linear streams, the space of programs themselves needs to become continuous. Neural networks represent a step in this direction. While the network topology itself is discrete (the connection between nodes is either present, or not), when one expresses the network topology via its weight-connectivity matrix, the degree to which any particular edge is present (the absolute value of the weight associated with it) can be made as small as desired, and this provides the continuity we are after.

The particular line of work we are presenting in this paper emerged in 2012–2013, when it was recognized by our group that the approximation domains providing continuous denotational semantics in the theory of programming languages can acquire the structure of vectors spaces, when equipped with cancellation properties missing in the standard theory of interval numbers. Namely, there must be enough overdefined (partially inconsistent) elements in those spaces to produce zero on addition by the mechanism of cancellation with underdefined (partially defined) elements. For interval numbers, this corresponds to introduction of pseudosegments $[a, b]$ with the contradictory property that $b < a$, following Warmus (1956). For probabilistic spaces, this corresponds to allowing negative values of probabilities in addition to usual non-negative values. The mathematics of the resulting *partial inconsistency landscape* is presented in section 4 of Bukatin & Matthews (2015c).

By 2015 it became apparent to our group that programming with linear streams was a rich formalism which included programming with streams of probabilistic samples and programming with *generalized animations*. This framework seemed to provide methods for *continuous higher-order programming*, and, moreover, it had good potential for obtaining more efficient schemas for genetic programming by allowing to introduce the motives similar to *regulation of gene expression* into genetic programming frameworks (Bukatin & Matthews 2015c). Crucially, it also became apparent at that time that if one introduced the discipline of interleaving linear and non-linear transformations of linear streams, then one could parametrize large classes of programs by matrices of numbers (Bukatin & Matthews 2015b;c).

The first open-source software prototypes associated with this approach also appeared in 2015 (Fluid 2015–2017).

In 2016 we understood that the resulting formalism generalized recurrent neural networks, and the term *dataflow matrix machines* was coined (Bukatin et al. 2016a). The modern version of the self-referential mechanism in DMMs and the first precise description of how dataflow matrix machines function, given that their matrices can be dynamically expanded, appeared in Bukatin et al. (2016b). The programming patterns for the resulting software framework were studied in Bukatin et al. (2016d). A more theoretical paper (Bukatin et al. 2016c) explored the possibility of using self-referential matrix transformations instead of lambda-calculus as the foundation in this context and established further connections between neural networks and the mathematics of the partial inconsistency landscape.

The formalism of finite prefix trees with numerical leaves (the vector space of recurrent maps) was introduced in the Fall of 2016. This formalism was inspired by our work with Clojure programming language (Hickey 2018). The first open-source implementation of a version of DMMs based on that formalism and written in Clojure was produced in that time frame (DMM 2016–2017), and a research paper based on this architecture was presented recently at LearnAut 2017 (Bukatin & Anthony 2017).

Acknowledgements

We would like to thank Dima-David Datjko, Elena Machkasova, and Elena Nekludova for helpful discussions of the material presented in Bukatin & Anthony (2017) and here.

References

Ba, J., G. Hinton, V. Mnih, J. Leibo and C. Ionescu. 2016. Using fast weights to attend to the recent past. October 2016. https://arxiv.org/abs/1610.06258

Bukatin, M. and J. Anthony. 2017. Dataflow matrix machines as a model of computations with linear streams. LearnAut 2017 ("Learning and Automata" Workshop at LICS 2017). https://arxiv.org/abs/1706.00648

Bukatin, M. and S. Matthews. 2015a. Almost continuous transformations of software and higher-order dataflow programming. July 2015. https://arxiv.org/abs/1601.01050

Bukatin, M. and S. Matthews. 2015b. Dataflow graphs as matrices and programming with higher-order matrix elements. August 2015. https://arxiv.org/abs/1601.01050

Bukatin, M. and S. Matthews. 2015c. Linear models of computation and program learning. In G. Gottlob, G. Sutcliffe and A. Voronkov (eds.) GCAI 2015. Global Conference on Artificial Intelligence, vol. 36. 66–78. https://easychair.org/publications/paper/Q4lW

Bukatin, M., S. Matthews and A. Radul. 2016a. Dataflow matrix machines as a generalization of recurrent neural networks. March 2016. http://arxiv.org/abs/1603.09002

Bukatin, M., S. Matthews and A. Radul. 2016b. Dataflow matrix machines as programmable, dynamically expandable, self-referential generalized recurrent neural networks. May 2016. https://arxiv.org/abs/1605.05296

Bukatin, M., S. Matthews and A. Radul. 2016c. Notes on pure dataflow matrix machines: Programming with self-referential matrix transformations. October 2016. https://arxiv.org/abs/1610.00831

Bukatin, M., S. Matthews and A. Radul. 2016d. Programming patterns in dataflow matrix machines and generalized recurrent neural nets. June 2016. https://arxiv.org/abs/1606.09470

DMM. 2016–2017. DMM project GitHub repository. https://github.com/jsa-aerial/dmm

Farnell, A. 2010. Designing sound. Cambridge, MA: MIT Press.

Fluid, X. 2015–2017. Project "Fluid" GitHub repository. https://github.com/anhinga/fluid

Graves, A., G. Wayne and I. Danihelka. 2014. Neural Turing machines. October 2014. https://arxiv.org/abs/1410.5401

He, K., X. Zhang, S. Ren and J. Sun. 2016. Identity mappings in Deep Residual Networks. March 2016. https://arxiv.org/abs/1603.05027

Hickey, R. 2008. The Clojure programming language. In Proceedings of the 2008 Symposium on Dynamic Languages. Association for Computing Machinery.

Hochreiter, S. and J. Schmidhuber. 1997. Long short-term memory. Neural Computation 9. 1735–1780.

Johnston, W., J. Hanna and R. Millar. 2004. Advances in dataflow programming languages. ACM Computing Surveys 36. 1–34.

Karpathy, A. 2015. The unreasonable effectiveness of Recurrent Neural Networks. May 2015. http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Kornai, A. 2008. Mathematical linguistics. London: Springer.

Mathews, M. 1963. The digital computer as a musical instrument. Science 142. 553–557.

McCulloch, W. and W. Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics 5. 115–133.

Merz, E. 2011. Sonifying processing: The Beads tutorial. http://evanxmerz.com/?page_id=18

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani and K. Q. Weinberger (eds.) Advances in neural information processing systems 26. Red Hook, NY: Curran Associates. 3111–3119

Murali, V., S. Chaudhuri and C. Jermaine. 2017. Bayesian sketch learning for program synthesis. March 2017. https://arxiv.org/abs/1703.05698

Nejati, A. 2016. Differentiable programming. August 2016. https://pseudoprofound.wordpress.com/2016/08/03/differentiable-programming.

Olah, C. 2015. Neural networks, types, and functional programming. September 2015. http://colah.github.io/posts/2015-09-NN-Types-FP.

Pollack, J. 1987. On connectionist models of natural language processing. Doctoral dissertation. University of Illinois at Urbana-Champaign.

Schmidhuber, J. 1993. A "self-referential" weight matrix. In S. Gielen and B. Kappen (eds.) ICANN 93: Proceedings of the International Conference on Artificial Neural Networks. Berlin & New York: Springer. 446–450.

Siegelmann, H. and E. Sontag. 1995. On the computational power of neural nets. Journal of Computer and System Sciences 50. 132–150.

Warmus, M. 1956. Calculus of approximations. Bulletin of the Polish Academy of Sciences Cl III 4. 253–259.

Weston, J., S. Chopra and A. Bordes. 2014. Memory networks. October 2014. https://arxiv.org/abs/1410.3916

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 187–217

A survey of proof nets and matrices for substructural logics

SEAN A. FULOP
California State University, Fresno
sfulop@csufresno.edu

KEYWORDS

linear logic
proof nets
matrix method
Lambek calculus
substructural logic

ABSTRACT

This paper is a survey of two kinds of "compressed" proof schemes, the *matrix method* and *proof nets*, as applied to a variety of logics ranging along the substructural hierarchy from classical all the way down to the nonassociative Lambek system. A novel treatment of proof nets for the latter is provided. Descriptions of proof nets and matrices are given in a uniform notation based on sequents, so that the properties of the schemes for the various logics can be easily compared.

1. Introduction

This paper provides a survey of two kinds of "compressed" proof schemes, the *matrix method* and *proof nets*, as applied to a variety of logics ranging along the substructural hierarchy (Restall 2000) from classical all the way down to the nonassociative Lambek system. There appears to be a paucity of survey literature that discusses proof nets for a variety of logics in a uniform notation, and even less which discusses matrix methods in relation to proof nets. It is the author's hope that this paper can provide in one source a host of information and methodology for proof nets and matrices, which could allow further research extending and using these techniques to be more easily conducted. There are few new results presented here, but the available literature provides treatments of various logics which are incommensurate; we hope to rectify this situation by unifying the presentation to a common framework.

Section 2 provides the necessary background, reviewing Gentzen-type sequent calculi for a variety of logics. Section 3 introduces the approaches to "proof compression" which are the main subject of the paper. Section 4 presents the matrix method, which works for both classical and linear logic, in some detail. As background, we also rehearse Smullyan's "unifying notation" (Smullyan 1963) of signed formulae which is central to the present paper. Section 5 describes proof nets for a variety of logics, beginning with the canonical case of multiplicative linear logic (Girard 1987). The proof nets are defined in a two-sided framework that can be directly applied to two-sided sequents, so that proof nets for the various logics can be readily compared. From here it is possible to go both up and down the substructural hierarchy; after also considering proof nets for classical logic, two versions of the Lambek calculus are treated. It is observed how the Lambek systems, with their increasingly rigid structural requirements on the layout of the formulae in a sequent, require more strictly geometrical conditions on correct proof nets. The nonassociative Lambek calculus is here provided with a two-sided proof net system and a geometric correctness condition for the first time. The last sections of the paper briefly discuss complexity of proof procedures and the general problem of "identity of proofs".

2. Sequent systems

2.1. Classical propositional logic

We begin the discussion with Gentzen's Gentzen (1934) sequent calculus for classical logic. This deductive system permits the proof of special *sequent* statements of the form $\Gamma \Rightarrow \Delta$. In a typical notation, $A, B, \ldots$ stand for proposition occurrences, while $\Theta, \Gamma, \ldots$ stand for sequences of proposition occurrences. A sequent in classical logic is often interpreted metalogically as a statement that the (possibly empty) formula sequence Δ, the *succedent,* follows from the (possibly empty) formula sequence Γ, the *antecedent,* in a natural deduction or axiomatic system of classical logic. To permit this interpretation, the succedent must be understood as a disjunction of its formulae, while the antecedent must be understood as a conjunction of its formulae. The standard (after Gentzen) presentation of the classical sequent calculus involves sequents, as just described, which may have formulae on either side of the arrow; such a presentation is known as a *two-sided* sequent calculus. Gentzen's rules of inference for the classical sequent calculus may be presented as follows:

Definition 1 (Classical sequent calculus, Gentzen 1934).

$D \Rightarrow D$ (Axiom)

$$\frac{A, \Gamma \Rightarrow \Theta, B}{\Gamma \Rightarrow \Theta, A \to B} \ (\to \text{R})$$

$$\frac{\Gamma \Rightarrow \Theta, A}{\neg A, \Gamma \Rightarrow \Theta} \ (\neg \text{L})$$

$$\frac{A, \Gamma \Rightarrow \Theta}{A \wedge B, \Gamma \Rightarrow \Theta} \quad \frac{B, \Gamma \Rightarrow \Theta}{A \wedge B, \Gamma \Rightarrow \Theta} \ (\wedge \text{L})$$

$$\frac{A, \Gamma \Rightarrow \Theta}{\Gamma \Rightarrow \Theta, \neg A} \ (\neg \text{R})$$

$$\frac{\Gamma \Rightarrow \Theta, A \quad \Gamma \Rightarrow \Theta, B}{\Gamma \Rightarrow \Theta, A \wedge B} \ (\wedge \text{R})$$

$$\frac{\Gamma \Rightarrow \Theta, A \quad B, \Delta \Rightarrow \Lambda}{A \to B, \Gamma, \Delta \Rightarrow \Theta, \Lambda} \ (\to \text{L})$$

$$\frac{A, \Gamma \Rightarrow \Theta \quad B, \Gamma \Rightarrow \Theta}{A \vee B, \Gamma \Rightarrow \Theta} \ (\vee \text{L})$$

$$\frac{\Gamma \Rightarrow \Theta, A}{\Gamma \Rightarrow \Theta, A \vee B} \quad \frac{\Gamma \Rightarrow \Theta, B}{\Gamma \Rightarrow \Theta, A \vee B} \ (\vee \text{R})$$

The above gives the so-called *logical rules* which show how the operators work. Because the left and right sides of a sequent are considered as sequences, to obtain classical logic one also requires Gentzen's *structural rules* – which are no less logical, in spite of the terms.

Definition 2 (Structural rules for the left side).

(Thinning)
$$\frac{\Gamma \Rightarrow \Theta}{\Gamma, A \Rightarrow \Theta}$$

(Exchange)
$$\frac{\Gamma_1, B, C, \Gamma_2 \Rightarrow \Theta}{\Gamma_1, C, B, \Gamma_2 \Rightarrow \Theta}$$

(Contraction)
$$\frac{\Gamma, A, A \Rightarrow \Theta}{\Gamma, A \Rightarrow \Theta}$$

There are mirror-image structural rules for the right side of sequents also, which are omitted for space reasons here. A sequent calculus proof is then a tree-like figure with initial sequents (possibly axioms) at the top and a conclusion at the bottom called the *endsequent.* To prove a single formula of classical logic, the initial sequents must be axioms, and the endsequent must have this formula as the succedent with an empty antecedent.

Some variations of the sequent calculus have been defined in which the antecedent and succedent are sets rather than sequences of formulae (e.g., Wallen 1990); in such a presentation the structural rules are "compiled in" to the definition of a sequent, and are not explicitly stated or used.

The only other rule permitted in a sequent calculus is known as "Cut", which certifies a kind of transitivity for sequents:

$$\frac{\Gamma \Rightarrow \Theta, D \qquad D, \Delta \Rightarrow \Lambda}{\Gamma, \Delta \Rightarrow \Theta, \Lambda}$$

Gentzen's important result was his "Hauptsatz" stating that the Cut rule can be eliminated with no loss of logical power for the system. The resulting Cut-free sequent system then enjoys the *subformula property,* meaning that "the formulae occurring in any [Cut-free] proof of a given endsequent are all subformulae of the endsequent" (Wallen 1990; using an obvious definition of *subformula*). It is plain from inspecting the Cut rule that this cannot be a property of proofs which use it. Thanks to the subformula property, a classical sequent proof search can be "goal-directed", working upward from the endsequent whose proof we seek by applying the inference rules in reverse. Of course, only Cut-free proofs can ever be discovered in this fashion.

A goal-directed deduction system is often called *analytic,* highlighting the idea that one analyzes the goal sequent to construct (or fail to construct) its proof. The opposite of this is then called a *synthetic* system, in which one works from the premisses to the proven expression. A natural deduction system (e.g., Gentzen 1934) is one example which is usually thought of as synthetic, since it is not generally used to construct a natural deduction proof "upwards" from the conclusion. It is worth noting, however, that natural deductions in the logics considered here can normally enjoy the subformula property, and so natural deduction can be regarded as closer to an analytic system than it is sometimes given credit for being. All of the proof methods discussed in this paper are analytic because our focus is on "compressed" goal-directed proof schemes, and so the logical systems to be discussed will be limited to their Cut-free versions. It is somewhat ironic that, as a referee pointed out, cut-free proofs are generally longer than proofs with cuts, so in one sense the compression of proofs is more difficult in the mathematically more pleasant realm of goal-directed theorem proving.

Smullyan (1968) developed a classical deductive system called *analytic tableaux* based upon foundations laid by Beth (1959). It is definitionally equivalent to a Cut-free sequent calculus handling sets of formulae (thus doing without structural rules), but the inference rules are explicitly turned upside-down, highlighting the analytic approach by placing the desired goal expression at the top of the proof (called a tableau).

2.2. Multiplicative linear logic

Linear logic was introduced by Girard (1987), and has since been the object of much study. For our discussion of compressed proof objects, only selected fragments will be used. We present a two-sided sequent formulation of the *unit-free* multiplicative fragment. It is two-sided in the previously used sense that the derivable sequents have possibly nonempty antecedent and succedent. This logic is commonly named MLL$^-$.

Definition 3 (Sequent calculus for MLL$^-$ Moot 2002)**.**

$$D \Rightarrow D \quad \text{(Axiom)} \qquad \frac{\Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow A^\perp, \Delta}\ ({}^\perp\ \mathrm{R})$$

$$\frac{\Gamma \Rightarrow A, \Delta}{\Gamma, A^\perp \Rightarrow \Delta}\ ({}^\perp\ \mathrm{L}) \qquad \frac{\Gamma, A, B \Rightarrow \Delta}{\Gamma, A \otimes B \Rightarrow \Delta}\ (\otimes\ \mathrm{L})$$

$$\frac{\Gamma \Rightarrow A, \Delta \quad \Gamma' \Rightarrow B, \Delta'}{\Gamma, \Gamma' \Rightarrow A \otimes B, \Delta, \Delta'}\ (\otimes\ \mathrm{R}) \qquad \frac{\Gamma \Rightarrow A, B, \Delta}{\Gamma \Rightarrow A \parr B, \Delta}\ (\parr\ \mathrm{R})$$

$$\frac{\Gamma, A \Rightarrow \Delta \quad \Gamma', B \Rightarrow \Delta'}{\Gamma, \Gamma', A \parr B \Rightarrow \Delta, \Delta'}\ (\parr\ \mathrm{L}) \qquad \frac{\Gamma \Rightarrow A, \Delta \quad \Gamma', B \Rightarrow \Delta'}{\Gamma, \Gamma', A \multimap B \Rightarrow \Delta, \Delta'}\ (\multimap\ \mathrm{L})$$

$$\frac{\Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \multimap B, \Delta}\ (\multimap\ \mathrm{R})$$

Linear logic is *substructural,* which means that some of Gentzen's structural rules for classical logic are not allowed. The only one of Gentzen's structural rules that is allowed now is Exchange, so the sequents in essence keep track of formula occurrences, meaning each side of the sequent constitutes effectively a multiset of occurrences. We also have the Cut rule allowed, but it is eliminable as before, and we focus only on the Cut-free version. There follows an example proof of a sequent in MLL$^-$:

$$\cfrac{\cfrac{\cfrac{\cfrac{A \Rightarrow A \quad B \Rightarrow B}{A \parr B \Rightarrow A, B}\ (\parr\ \mathrm{R}) \qquad C \Rightarrow C}{A \parr B, C \Rightarrow A, B \otimes C}\ (\otimes\ \mathrm{R})}{A \parr B \Rightarrow C \multimap A, B \otimes C}\ (\multimap\ \mathrm{R})}{A \parr B, (B \otimes C)^{\perp} \Rightarrow C \multimap A}\ ({}^{\perp}\ \mathrm{L})$$

2.3. Associative Lambek calculus

Now, we discuss other substructural logics which take away more of the structural rules, both explicit and implicit. An important motivation for these logics is found in linguistics, where they serve as fundamental systems within theories of "categorial grammar" and its extension to "type-logical grammar" (Morrill 1994; Fulop 2004). Our first example is a logic that was first introduced as a "syntactic calculus" operating on formulae that were interpreted as linguistic parts of speech (Lambek 1958). In this guise it is known as the (associative) Lambek calculus.

Definition 4 (Lambek sequent calculus, Lambek 1958)**.**

$$D \Rightarrow D \quad (\mathrm{Axiom}) \qquad\qquad \frac{\Delta \Rightarrow B \quad \Gamma[A] \Rightarrow C}{\Gamma[\Delta, B\backslash A] \Rightarrow C}\ (\backslash \mathrm{L})$$

$$\frac{\Delta \Rightarrow B \quad \Gamma[A] \Rightarrow C}{\Gamma[A/B, \Delta] \Rightarrow C}\ (/\mathrm{L}) \qquad\qquad \frac{B, \Gamma \Rightarrow A}{\Gamma \Rightarrow B\backslash A}\ (\backslash \mathrm{R})$$

$$\frac{\Gamma, B \Rightarrow A}{\Gamma \Rightarrow A/B}\ (/\mathrm{R}) \qquad\qquad \frac{\Gamma[A, B] \Rightarrow C}{\Gamma[A \bullet B] \Rightarrow C}\ (\bullet \mathrm{L})$$

$$\frac{\Gamma \Rightarrow A \quad \Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow A \bullet B}\ (\bullet \mathrm{R}) \qquad\qquad \frac{\Delta \Rightarrow A \quad \Gamma[A] \Rightarrow C}{\Gamma[\Delta] \Rightarrow C}\ (\mathrm{Cut})$$

Lambek calculus (notated simply L, or L_{ϵ} when empty antecedents are permitted) is a positive logic in which none of Weakening, Contraction, or Exchange are permitted, so the logical consequence relation involves sequences of formulae. Thus we introduced the standard notation $\Gamma[\cdot]$, which means a formula sequence with a place identified for substitution which is matched by another use of the similar notation in the same inference rule. Once again, the logic enjoys Cut elimination so we deal solely with the Cut-free version. Associativity of the sequences is assumed, as a kind of implicit structural rule; we show what happens next when even this is removed.

2.4. Nonassociative Lambek calculus

The last system to be introduced is a version of Lambek calculus from which even the implicit structural rule of associativity is taken away. This nonassociative Lambek system NL was first described in 1961 (Lambek 1961) where it was motivated by linguistic applications, and it has since been recognized as fundamental within the area of type-logical grammars for linguistics (Moortgat 1997; Fulop 2004). This system is especially useful for grammatical deductions because, without associative sequences, the sequent calculus handles binary trees of formulae that can be used to represent the syntactic structures of languages. The sequent presentation below does without the product operator '•', because this is logically superfluous in a sequent formulation (as it is even in the associative system L above). The nonassociativity of the sequents is here emphasized by replacing the usual comma with the sequent-level operator '⋄'. The sequent system enjoys Cut-elimination and is single-conclusion, so that all provable sequents have a single succedent formula.

Definition 5 (NL sequent calculus, Lambek 1961)**.**

$$A \Rightarrow A \quad \text{(Axiom)} \qquad \frac{\Delta \Rightarrow B \quad \Gamma[A] \Rightarrow C}{\Gamma[(\Delta \diamond B\backslash A)] \Rightarrow C} \ (\backslash\ \mathrm{L})$$

$$\frac{\Delta \Rightarrow A \quad \Gamma[A] \Rightarrow C}{\Gamma[\Delta] \Rightarrow C} \ (\mathrm{Cut}) \qquad \frac{(\Gamma \diamond B) \Rightarrow A}{\Gamma \Rightarrow A/B} \ (/\ \mathrm{R})$$

$$\frac{\Delta \Rightarrow B \quad \Gamma[A] \Rightarrow C}{\Gamma[(A/B \diamond \Delta)] \Rightarrow C} \ (/\ \mathrm{L}) \qquad \frac{(B \diamond \Gamma) \Rightarrow A}{\Gamma \Rightarrow B\backslash A} \ (\backslash\ \mathrm{R})$$

3. Proof compression

A key application of analytic deductive methods has been automated deduction. A significant problem for the sequent/tableau systems in this arena is the inefficiency resulting from a large search space. Much progress has been made in the development of efficient proof search by application of goal-directed logic programming methods such as resolution (e.g., Gabbay & Olivetti 2000). The primary focus here will not be on efficient search for complete proofs, but rather on the question of how can an analytic proof be compressed, and thereby become a fundamentally different sort of object that can be viewed in a new way, and possibly constructed more efficiently.

It has been explained many times in the literature (e.g., Dyckhoff & Pinto 1999) that, even restricting attention to Cut-free proofs, the sequent and tableau calculi may validate numerous proofs of a given sequent. These several proofs may differ either "trivially" or non-trivially in the order of application of some of the rules. The propensity for the sequent/tableau systems to suffer from *spurious ambiguity* caused by trivial rule permutation has been explained in detail elsewhere (Wallen 1990), so here we simply note the fact and consider its ramifications and proposed remedies. In this paper, we will consider two kinds of "compressed proof objects", which differ philosophically with respect to the spurious ambiguities just mentioned. The first of these, the *matrix method,* constructs a minimal compressed proof object that is really nothing beyond a provability test. There is a compelling argument that a matrix proof of a sequent is not really a proof anymore, because not only trivially different, but also nontrivially different sequent/tableau proofs are all collapsed. A provable sequent has, by definition, exactly one matrix demonstrating its validity.

The second kind of compressed proof object to be considered here is the *proof net.* Proof nets were originally described for linear logic (Girard 1987), and it has been claimed for that system that they compress proofs to "just the right extent", in the sense that any two sequent/tableau proofs which are *nontrivially different* will have distinct corresponding proof nets, while any two sequent/tableau proofs which differ only *spuriously* (i.e., by "harmless" permutations of the rules) will have the same proof net corresponding (Straßburger 2006). The philosophy behind proof nets is to always seek, if not achieve, this correspondence for the logic at hand, because a proof net is supposed to be something beyond a minimal provability test – proponents think of it as "the essence of a proof." There is, however, no current consensus among logicians as to what precisely should count as a *nontrivial* difference between two proofs.

4. Matrix methods

In classical and intuitionistic logic, the redundancies and other difficulties with standard proof calculi led to the matrix methods, independently invented by Bibel (1980) and Andrews (1981), but perfected by Wallen (1990). Here we follow Wallen's exposition, and the unifying notation of Smullyan's signed formulae will be of utmost importance.

4.1. Unifying notation of signed formulae

It would be redundant to present the inference rules of Smullyan's tableau calculus for classical logic, since they are essentially the same as those of Gentzen's sequent calculus. One important element of Smullyan's treatment that will be important for our discussion throughout, however, is his "unifying notation" which uses *signed formulae* classified into two basic varieties Smullyan (1963). A signed formula is just a formula P which is annotated by a sign, or *polarity,* which we will show as either P^+ or P^-. The sign is used to indicate the "negation environment" of the formula occurrence within a sequent, so that negative formula occurrences are all those that are within the scope of an explicit or implicit negation. An "implicit negation" environment is always (and only) the antecedent of a conditional or of a sequent. This definition comes from the truth-functional equivalence between formulae $A \to B$ and $\neg A \vee B$. Smullyan used signed formulae to avoid writing sequents directly with the sequent arrow; his rules for positive formulae exactly mirror the succedent (R) rules in the sequent calculus, while tableau rules for negative formulae mirror antecedent (L) rules in the sequent calculus.

Signed formulae are then classified by Smullyan into two fundamental kinds, which can be determined by inspecting the sequent rules shown above. The key question is whether the inference rule "branches", having two premisses. A branching rule governs a "signed formula of type B", which we may call *disjunctive,* after the canonical example of the rule ($\vee$ L). A rule with only one premiss, on the other hand, governs a "signed formula of type A", which we may term *conjunctive.* The conjunctive signed formulae in classical logic are these:

$$(A \wedge B)^-, (A \vee B)^+, (A \to B)^+, (\neg A)^+, (\neg A)^-$$

The disjunctive formulae are the others:

$$(A \vee B)^-, (A \wedge B)^+, (A \to B)^-$$

4.2. Classical logic matrices

Step one of the matrix method, and also ultimately of the proof net method, is to decompose the target sequent or formula into a tree of its subformulae that keeps track of the signs.

Definition 6 (Wallen 1990). *A* formula tree *for a signed formula $A^g, g \in \{+, -\}$ is a tree of subformulae of A together with an assignment of a sign to each position k of the formula tree. Each position k then contains a signed formula* occurrence *from within A; the formula occurrence apart from its sign at k is called the* label *of k. Let* $\mathrm{lab}(k)$ *and* $\mathrm{sgn}(k)$ *denote the label and sign of position k respectively. For each position k, if* $\mathrm{lab}(k)$ *occurs positively in A^g, then* $\mathrm{sgn}(k) = g$. *If* $\mathrm{lab}(k)$ *occurs negatively in A^g, then* $\mathrm{sgn}(k)$ *is the opposite sign from g.*

Definition 7 (Wallen 1990). *A* path *through formula A^g is a subset of the positions of its formula tree, defined inductively:*

1. *$\{k_0\}$, the root position, is a path.*
2. *If s, α is a path, so is $(s - \{\alpha\}), \alpha_1, \alpha_2$, for conjunctive α having α_1, α_2 as immediate subformulae.*
3. *If s, α is a path, so is $(s - \{\alpha\}), \alpha_1$, for conjunctive α having α_1 as its sole immediate subformulae (this is the case where α is a negation).*
4. *If s, β is a path, so is $(s - \{\beta\}), \beta_1$, and so is $(s - \{\beta\}), \beta_2$, for disjunctive β having β_1, β_2 as immediate subformulae.*

The second through fourth clauses above can be regarded as path reduction *steps. A completely reduced path will consist entirely of (signed) positions labeled by atoms, and is called an* atomic *path.*

The above formulation can be easily extended from signed formulae to two-sided sequents of signed formulae. The simplest way is to decompose each of the antecedent formulae and succedent formulae separately. The antecedent formulae are negatively signed, while the succedent formulae are positively signed, and one decomposes the whole set of signed formulae into a set of formula trees as above, treating the compound tree as a single graph with "multiple roots." The above definitions of a path through the tree and an atomic path can then be applied *mutatis mutandis.*

Definition 8 (Matrix representation Wallen 1990).

1. *If signed formula A^g is conjunctive, its matrix representation is a row matrix having as element(s) the signed component(s) found immediately below in its formula tree.*
2. *If signed formula A^g is disjunctive, its matrix representation is a column matrix having as elements the signed components found immediately below in its formula tree.*

3. If signed formula A^g is atomic, it is its own matrix representation.

A completed matrix for a signed formula must have every subformula in every submatrix converted to matrix representation; matrices are to be nested as needed. The matrix representation extends to signed two-sided classical sequents by a simple adaptation of the procedure described above for sequent trees. The matrix of a sequent is then simply a “row matrix” whose elements are the respective matrices of the constituent formulae. This fact can be related to the “metalogical” view of a sequent in which the antecedent formulae are conjoined while the succedent formulae are disjoined; observe that a conjunction in the antecedent and a disjunction in the succedent are each formulae of conjunctive type, and so a row matrix is the correct form for each.

Every atomic path through a signed formula (or sequent) is now represented by the sequence of signed atoms encountered on a left-right sequence of steps through the columns in its completed matrix – submatrices are to be stepped through as well in this procedure Wallen (1990). When a step enters a column matrix, only one row is selected (nondeterministically) for the atomic path, while the other is ignored.

The key idea at the core of the matrix method is that of a spanning set of connections.

Definition 9. *A* connection *is a pair of atomic positions in some path through A^g, which have the same label but opposite signs. A given set of connections is said to* span *A^g iff every atomic path through A^g contains a connection from the set.*

Theorem 10 (Wallen 1990)**.** *For signed propositional formula A^+, the existence of a spanning set of connections for it ensures its provability in classical logic, and vice versa.*

The above definition and theorem concerning a spanning set of connections also extends in a simple fashion *mutatis mutandis* to signed two-sided sequents. For clarity, this may be stated as follows:

Corollary 11. *For signed sequent $\Gamma^- \Rightarrow \Delta^+$, the existence of a spanning set of connections for it ensures its provability in classical sequent calculus, and vice versa.*

An example sequent provable in classical logic is shown in (1); the corresponding matrix of the sequent is shown in (2).

$$\neg A,\ B \to A \Rightarrow \neg B \tag{1}$$

$$\begin{bmatrix} [A^+] & \begin{bmatrix} B^+ \\ A^- \end{bmatrix} \end{bmatrix} \quad [B^-] \tag{2}$$

This matrix presents two atomic paths: (A^+, B^+, B^-) and (A^+, A^-, B^-). The spanning set of connections is then $\{\langle A^+, A^-\rangle, \langle B^+, B^-\rangle\}$, the existence of which shows the sequent to be provable in classical logic. Any sequent of classical propositional logic can be tested for provability using our adaptation of Wallen's matrix method to two-sided sequents. The sequent calculus (or tableaux method) can now be regarded as extremely inefficient methods of checking that every atomic path through the goal sequent tree contains a connection from a spanning set!

4.3. MLL$^-$

The matrix method for linear logic was presented by Galmiche (2000). The matrix representation of a sequent is obtained from the signed sequent tree just as with classical logic. The matrix for the sequent proven above in subsection 2.2 is shown in (3); a spanning set of connections for this matrix is given in (4).

$$\begin{bmatrix} A^- \\ B^- \end{bmatrix} \quad \begin{bmatrix} B^+ \\ C^+ \end{bmatrix} \quad \begin{bmatrix} C^- & A^+ \end{bmatrix} \tag{3}$$

$$\{\langle A^+, A^-\rangle, \langle B^+, B^-\rangle, \langle C^+, C^-\rangle\} \tag{4}$$

Reflecting the differences between MLL$^-$ and classical logic, it is no longer sufficient that the sequent matrix possesses a spanning set of connections, however. Galmiche stated the additional requirements for the set $\mathcal{S}$ of connections to *linearly span* a matrix in the following way: (1) All atomic occurrences in the matrix occur exactly once with each polarity in $\mathcal{S}$; (2) no pair of connections in $\mathcal{S}$ has overlapping elements; (3) $\mathcal{S}$ is a minimal spanning set. It is easy to see that the set of connections in (4) does linearly span the above matrix for the sequent. Galmiche stated the theorem that a sequent of MLL$^-$ is provable just in case its matrix possesses a set of connections which linearly spans it. It is notable that, despite presenting this as a proven fact, Galmiche never really proved it in his paper.

It is nevertheless possible for us to appreciate, at a glance at least, how the additional conditions defining a linearly spanning set derive from the characteristic that MLL handles multisets of formulae (cf. condition 1), effectively keeping track of formula occurrences and not allowing contractions of repeated formulae (cf. condition 2) or extraneous formulae (cf. condition 3) into a proof.

Having discussed the matrix method and signed formulae, it will be much easier to understand proof nets, a subject to which we turn next.

5. Proof net methods

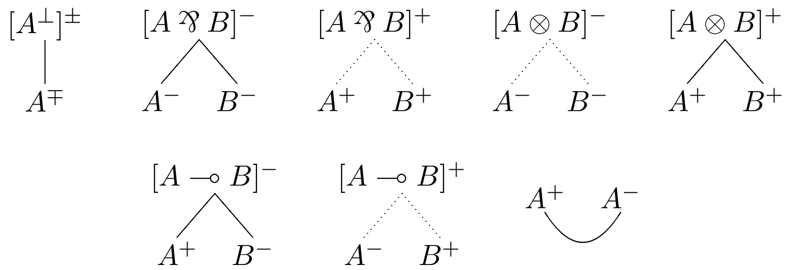
The sequent and tableau methods yield too many possible proofs of a given sequent, and have an undesirable amount of notational redundancy for automated theorem proving applications. The matrix method described above has certainly eliminated the redundancy, but now there are in a sense too few proofs of a given sequent for some applications; in fact, each provable sequent has precisely one matrix. This may be acceptable for theorem proving, but there are theoretical reasons to desire a proof representation that captures "the essence of a proof." This notion is related to the general problem of the identity of proofs (Došen 2003), and the compressed proof objects known as *proof nets* have been put forth as solving this problem for MLL, at least Straßburger (2006).

5.1. MLL^{-}

Our discussion of proof nets must begin with MLL^{-}, since Girard (1987) developed linear logic and proof nets at the same time. The proof nets are in fact a subclass of the decomposition graphs known as proof structures. Just as the matrix of a sequent is produced from the decomposition of the signed formulae while distinguishing conjunctive from disjunctive formulae, a *proof structure* is a special graph that is drawn from a formula or sequent decomposition, also keeping track of the polarities and the conjunctive or disjunctive nature of the signed formulae. The subgraphs which are drawn for each type of formula decomposition are traditionally called *links*; to complete the proof structure, pairs of atoms having opposite polarity are linked together by edges called *axiom links.*

The MLL^{-} link schemata which may be used to decompose formulae in a proof structure are shown below. It should be mentioned that these link drawings are to be viewed as representing graphs without further geometric properties, so that the specific orientation of a link drawing or whether edges cross is unimportant. A complete proof structure for the provable MLL sequent already studied above follows the link presentation below. The resulting graph has two sorts of edges, which serve to distinguish the conjunctive from the disjunctive binary formula occurrences (negation is excluded from the conjunctive/disjunctive dichotomy for this purpose). The binary links shown with solid lines are the disjunctive formulae, traditionally called *times* links (typified by the link for the times connective $\otimes$ in a positive context), while the dotted lines show the conjunctive formulae, traditionally called *par* links (and typified by the link for the par $\parr$ connective in a positive context). Axiom links are shown with curved lines in the proof structures. Some formal definitions follow.

MLL$^-$ links:

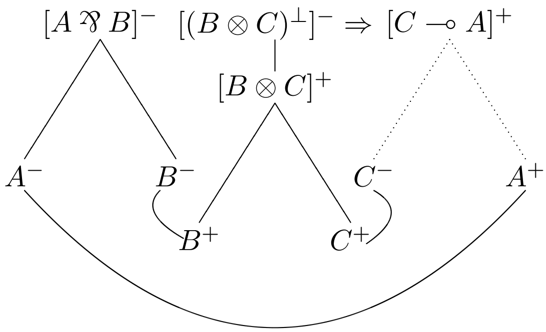


Formula(e) on top of each link are called *conclusion,* and formulae on the bottom of a link are called *premiss.* The axiom link is unique in having no premisses and two conclusions.

MLL$^-$ proof structure:

Definition 12 (Moot 2002)**.** *A* proof structure $\langle S, \mathcal{L}\rangle$ *consists of a set S of signed formulae and a set $\mathcal{L}$ of links over S (from the above possibilities). A proof structure must also satisfy the conditions:*

- *Every formula in S is at most once the premiss of a link;*
- *Every formula in S is exactly once the conclusion of a link.*



The proof structure exemplified above is *two-sided,* because it can be created for a sequent with formulae on both sides of the arrow ⇒. It is possible to enumerate all possible proof structures for any sequent now by decomposing all connectives until we reach the atomic formulae, and then connecting positive to negative atoms using axiom links (Moot 2002). The two-sided means of presenting a proof structure is, however, not common in literature on linear logic, and has never been fully described in published literature.[1] In the literature, MLL proof structures are almost invariably one-sided – meaning they can be constructed only for a one-sided sequent calculus with empty antecedents. Moreover, such structures cannot involve the implication or negation operators overtly as above, because they furthermore do not keep track of the polarities of formulae. For our purposes, the usual one-sided proof structures for MLL obscure the fundamental relationship with matrix methods and the unifying notation of signed formulae, so here we stick with the two-sided dialect.

Completing a proof structure for an MLL sequent is an important step toward demonstrating provability of the sequent, but it is not yet sufficient. A proof structure for a provable sequent is known as a *proof net,* and only those structures which satisfy an additional correctness condition are indeed proof nets. An impressive list of alternative correctness conditions for MLL proof nets has arisen from years of research on the topic, beginning with the original “long trip” condition of Girard (1987). This condition is somewhat cumbersome for our purpose here, so we will first describe a popular correctness condition due to Danos and Regnier (1989).

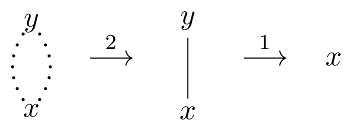
[1] The presentation here is derived from class lecture notes Straßburger (2006).

Take a proof structure as a graph; let us call it $\mathcal{P}$. Now, let $\sigma(\mathcal{P})$ be a new graph derived from $\mathcal{P}$ by deleting some edges. Specifically, σ acts to delete one edge nondeterministically from each par link in $\mathcal{P}$, and is called a *DR switching.*

Theorem 13 (Danos–Regnier correctness condition)**.** *A proof structure $\mathcal{P}$ is a proof net if and only if every DR switching $\sigma(\mathcal{P})$ of it yields a connected acyclic graph.*

The Danos–Regnier switching condition is easy to apply to small proof structures – the structure presented above is easily seen to satisfy it – but has exponential complexity because of the need to check the result of every DR switching of a proof structure for acyclicity and connectedness Moot (2002).

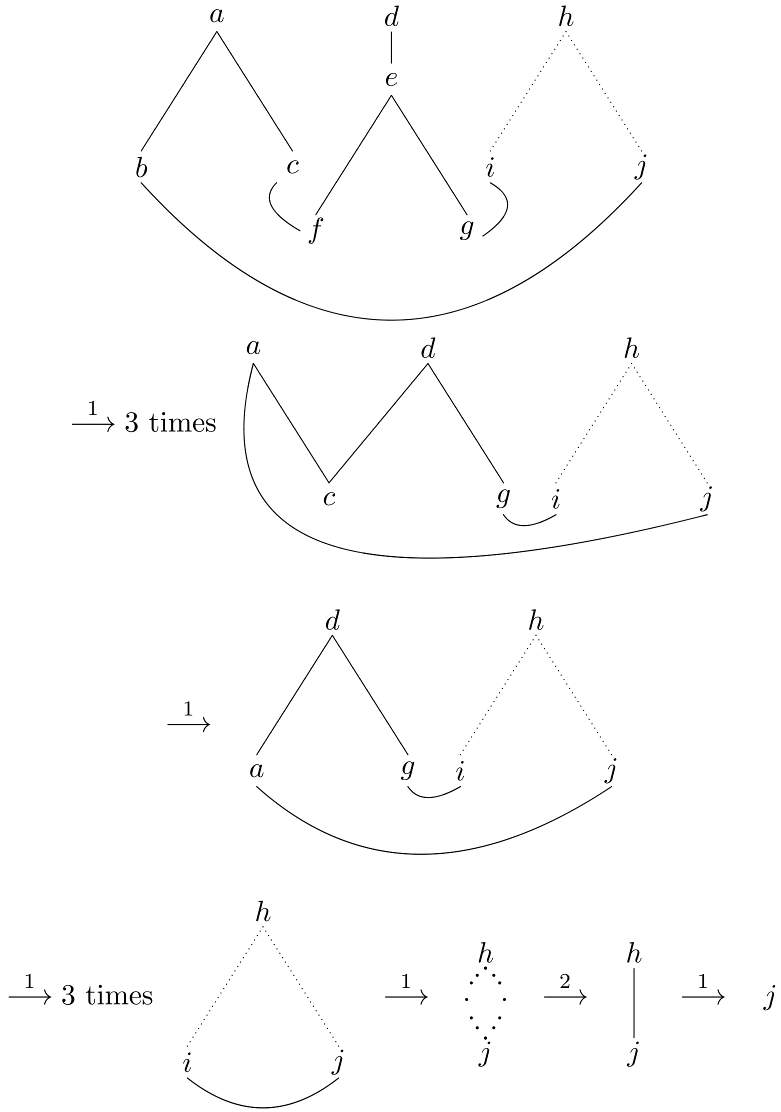
A more efficient condition was first presented in the PhD dissertation of Danos (1990), and involves transforming a candidate proof structure by graph contractions (v. Gross & Tucker 1987 for a formal definition of graph contraction). The two Danos contraction rules are presented as follows in Moot (2002):



The basic idea is that two ‘par’ edges transform to a single ‘times’ edge just if they connect the same two vertices (this can only result from previous contractions), and any two vertices linked by a ‘times’ edge contract to one vertex.

Theorem 14 (Danos contraction condition)**.** *A proof structure is a proof net if and only if it contracts to a single vertex by successive application of the Danos contraction rules above.*

The figure sequence below shows the successive contraction of the proof structure presented above; formulae are irrelevant for this condition, and are replaced by simple vertex labels. The equivalence between the Danos–Regnier switching condition and the Danos contraction condition was proven very simply in Straßburger (2006).


a d h
e
b c i j
f g
1
3 times
a d h
c g i j
1
d h
a g i j
1
3 times
h
i j
1
h
j
2
h
j
1
j

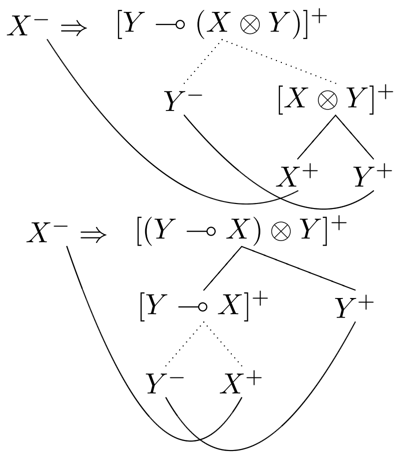
5.2. Intuitionistic MLL

We refer back to the sequent calculus rules for MLL^{-}; this system is rendered intuitionistic by endowing it with the single-conclusion property, by which all sequents must have just one succedent formula (Moot 2002). The positive fragment of this system with only $\otimes$ and $\multimap$ is known in the literature as multiplicative *intuitionistic* linear logic (MILL), and it has some thinly disguised early roots.

A kind of decomposition graph for MILL formulae was published by Kelly and Mac Lane in 1971 in their study of coherence in categories, and is possibly the first work on a compressed proof object showing aspects of the matrix and proof net methods. The Kelly–Mac Lane graph of a MILL formula shows its decomposition to signed atoms; if linking atoms in opposite polarity pairs can be achieved, then one has essentially a proof structure, but a correctness criterion is still required for such a structure to be a proof net (Moot 2002).

To build a proof structure in MILL, one begins as in MLL by decomposing the sequent into subformulae down to the atoms while keeping track of the polarities and the conjunctive/disjunctive property of the formula at each stage. The antecedent formulae are first given a negative sign while the succedent formula is given a positive sign. Beyond this there are just two operators, and the decomposition proceeds so that $[X \otimes Y]^{\pm}$ yields $X^{\pm}, Y^{\pm}$, while $[X \multimap Y]^{\pm}$ yields $X^{\mp}, Y^{\pm}$, as above in MLL. Signed formulae of the form $[X \otimes Y]^{-}, [X \multimap Y]^{+}$ are the conjunctive ones as in MLL, which are assigned a par link. The formal definition of a proof structure in MILL (without units) is the same as the above for MLL^{-} *mutatis mutandis*, and either of the above correctness conditions for MLL proof nets carries over to the case of MILL (Moot 2002).

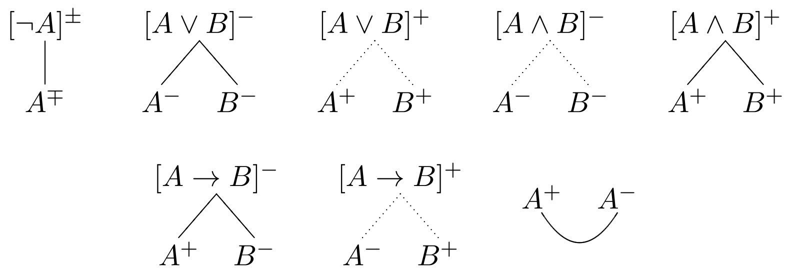
Below we show two proof structures for posited sequents of MILL; only the first one is a correct proof net, in which each DR switching yields a connected acyclic graph. The second proof structure has a cycle following removal of the right branch of the par link, demonstrating the posited sequent to be underivable in MILL. We see that MILL proof nets are merely a subspecies of MLL nets, however, one reason to discuss this logic separately here is to highlight the much earlier literature (Kelly & Mac Lane 1971) that first defined proof structures for this system, and was also first to make use of signed formulae in a linear logic system. MILL is in a sense also the archetypal logic in this family possessing the single-conclusion property.



5.3. Classical logic

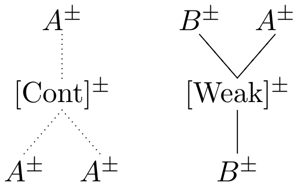
Classical (propositional) logic is actually quite similar to linear logic; all of the differences derive from the presence of Gentzen's structural rules of Weakening and Contraction. It is interesting to see how the definition of a proof net carries over to this case. Proof nets for classical logic have been developed by Robinson (2003) following the two-sided paradigm given above for linear logic, in which there are distinct links for decomposing each connective in a positive versus a negative context. The system therefore derives naturally from Smullyan's unifying notation for classical logic, although Robinson did not cite this prior literature. The conjunctive and disjunctive links for the decomposition of signed formulae are very similar to the ones needed for MLL, and are presented below with adjustments to suit our notation here (leaving aside the degenerate links which would handle the true and false units, not used here).

Classical logic links:



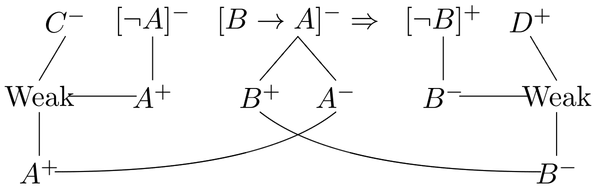
As Robinson showed, more is needed to obtain a kind of proof net that enjoys the same correctness conditions which govern MLL. Specifically, Robinson added links corresponding to the structural rules of Contraction and Weakening; the former are conjunctive while the latter are disjunctive. Once again our presentation changes the link notation somewhat to make it uniform with the presentation of MLL.

Structural links:



With these additions, a proof structure can be constructed for a classical sequent, relying on Definition 12 from the MLL case. The correctness conditions it must meet to be a proof net for a provable sequent are also carried over from MLL with no changes. An example is now shown.

Classical proof net:



This classical proof net turns out to have no conjunctive links, so it has to be a connected acyclic graph as it is shown according to the Danos–Regnier condition, and indeed it is. Robinson also gave an elegant, simple explanation connecting this correctness condition to the unifying notation, to be restated now. If a proof net comes from a proof, the graph must be simply connected, which forces the switching condition in the following way. A disjunctive ('times') signed subformula $A \circ B$ for any operator $\circ$ has "branched" premisses which come from separate subproofs, and so are not yet connected, so the occurrence of $A \circ B$ must be joined to both premisses otherwise the proof net would end up disconnected. On the other hand, a conjunctive ('par') signed subformula $C \circ D$ has premisses coming from the same subproof, so they are already connected. The formula $C \circ D$ must then be joined to exactly one premiss or the graph will contain a cycle. This explanation is also applicable to the linear logic cases. It is interesting that the only real difference between the MLL^- and classical proof net systems is the addition of the special links for Contraction and Weakening.

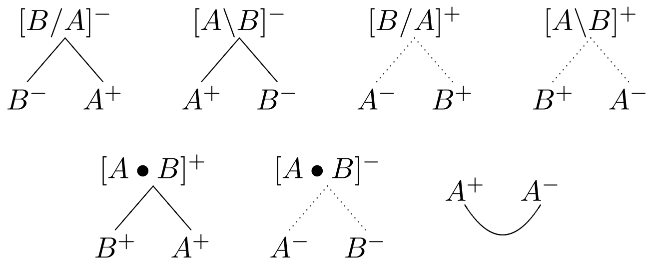
5.4. Associative Lambek calculus

So far we have discussed classical logic, which in essence handles formulae in sets, and two varieties of linear logic, which remove the Weakening and Contraction rules, and thereby keep track of occurrences of formulae. It is useful to note at this juncture that these logics have both matrix and proof net methods available for checking provability of sequents, neither of whose correctness conditions refer crucially to the geometrical arrangement of the proof object. It is apparent that a matrix of a sequent does not have any interesting geometrical properties; moreover, although a proof net is a kind of graph, there is nothing very "geometrical" about these proof nets so far. It is unimportant whether the link lines in a drawing of the graph cross, for example.

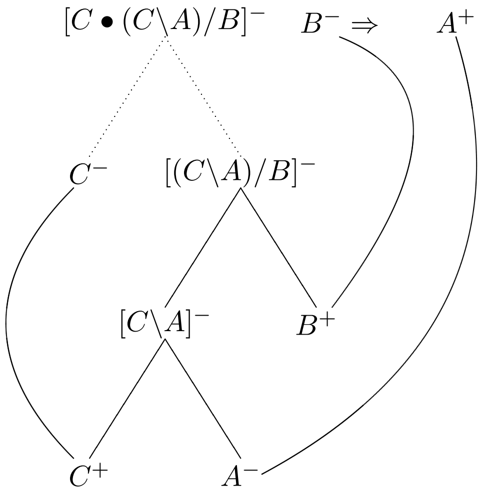
In fact, L_ϵ is basically MILL without Exchange. The lack of Exchange (or "commutativity") has effectively split the linear implication into a pair of directionally sensitive implications notated with the slash operators. Each slash is interpreted as saying that the formula on top of the slash results when the formula under it is adjacent on that side. The newfound sensitivity of the logic to the arrangement of formulae in a sequence has a profound effect on the definition of a compressed proof procedure. Below, the binary links for proof nets in the Lambek system L are provided, following Roorda (1991; 1992); this time, however, the geometry of the drawings as shown provides important information. The left-right arrangement of the subformulae in a decomposition link is now critical; one must swap the order of the subformulae with respect to the parent formula *in the positive links only.*[2]

[2] One of the very few sources to provide these link drawings for Lambek proof nets (Moot 2002) has got this condition backwards, unfortunately.

Links for L_ϵ:



L proof structure. It seems that the formal definition of a proof structure in L can be kept the same as for the systems above. An example is now shown.



This example is in fact also a proof net for the provable sequent. This proof net satisfies the Danos–Regnier condition plus an additional requirement of planarity which was first proven necessary by Roorda (1991); each DR-switching graph is not only acyclic connected, but also planar as shown in the drawing.

Although this treatment here applies generally to the system L_ϵ allowing empty antecedents, it has been shown (Lamarche & Retoré 1996) that we may exclude all sequents with empty antecedent by an additional requirement about *subnets* of a proof net. A subnet is, in our notation, a down-closed subset of the nodes such that axiom links stay inside the substructure. Then, to exclude sequents with empty antecedent, it is sufficient to require that every subnet of a proof net possess at least two conclusions (i.e., local root formulae at the top).

A different presentation of a noncommutative linear logic was also shown to require planar proof nets (Abrusci 1991), around the same time as Roorda's result about the Lambek system. The NCMLL system described in the reference is equivalent in its multiplicative fragment to another noncommutative logic (Pentus 1997), which in turn is a conservative extension of L_ϵ (Abrusci 1997). Thus it is beginning to look as if noncommutativity of the logic (i.e., lack of Exchange) leads directly to a new geometric requirement of planarity of the proof net. It is also not at all obvious that a version of the matrix method could somehow be formulated for this kind of logic, for now the specific arrangement of the formulae is critical.

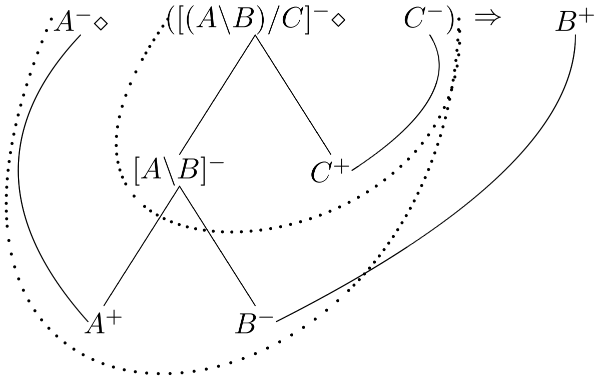
5.5. Nonassociative Lambek calculus

Despite having been discussed many times in the literature, the system NL has never had a proof net scheme defined for it in a way that relates clearly with the other proof net schemes presented here.[3] It turns out to be quite easy to adapt the proof nets for associative L, but an additional correctness condition is required that has never been developed in literature, and which makes the resulting nets even more "geometrical." This is the only novel result to be introduced in the present survey.

Let us discuss several examples of NL proof structures to develop the additional correctness condition. Example 1 shows the basic kind of structure for a provable sequent, which is planar just as in system L. A further correctness condition is needed in order to account for the effects of the parentheses, which govern the nonassociative structure of the antecedent. To develop this extra condition, we draw dotted boundaries from each pair of parentheses in the antecedent, extending around the first decomposition link whose active conclusion subformula is governed by that pair. Such boundaries in our proof nets will be called *parenthetical boundaries.* Examining the axiom links in the final structure, observe that only the link coming from the negative B atom, which connects to the succedent, crosses the parenthetical boundary that contains it.

[3] A proof net system for "classical" NL was provided in de Groote & Lamarche (2002), but these authors used a quite different formulation whose definition and correctness condition bears little obvious resemblance to the proof nets so far discussed.

Example 1



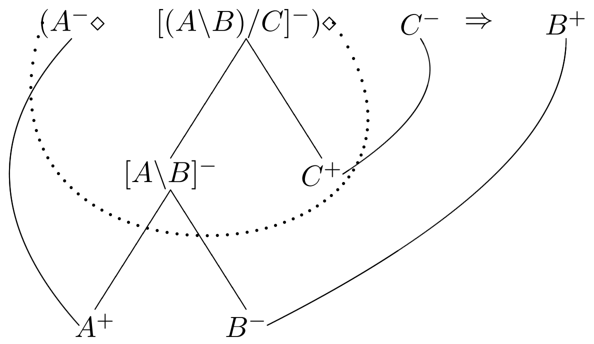
Example 2, by contrast, shows a similar proof structure for a non-provable sequent whose antecedent has the parentheses wrong. The structure is indeed planar, so the sequent would be provable in system L by invoking associativity, but observe that now both of the antecedent axiom links cross the first boundary (the outer boundary is not shown). The problem, in reality, is the C-link, because *the link from the positive atom crosses the parenthetical boundary which contains it.* We therefore state this as part of the correctness conditions.

Theorem 15. *An NL-proof structure is a proof net for the decomposed sequent just in case:*

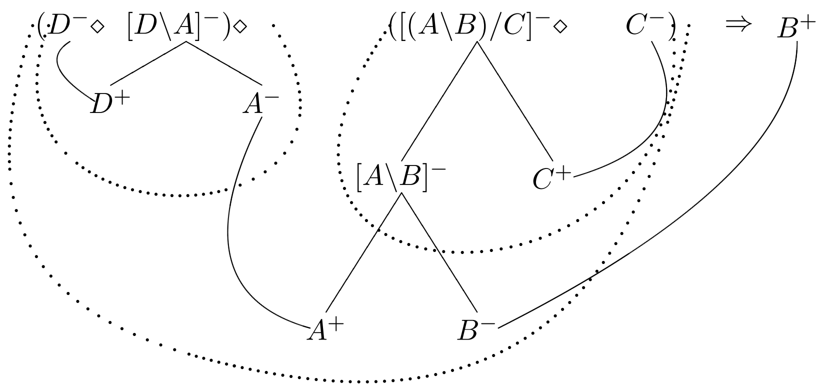
- *the Danos–Regnier switching condition, or other equivalent condition, holds of the structure;*
- *the structure is planar;*
- *no axiom link from a positive atom crosses the parenthetical boundary which contains it.*

Example 3 illustrates the structure for a more complicated provable sequent, and we observe that two axiom links cross boundaries, but neither involves a link from a positive atom crossing the boundary which contains it.

Example 2



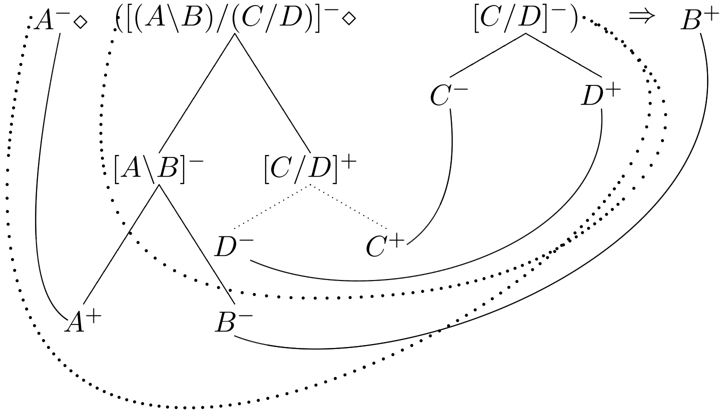
Example 3



It is quite easy to see the necessity of this correctness condition, so a brief explanation should suffice here. Note first that for each slash operator in a provable NL sequent, there must be a pair of parentheses surrounding a formula which contains it, and also surrounding the neighboring occurrence of the subformula under the slash. Every atomic subformula under a top-level slash (i.e., one not itself within a proper subformula) in the antecedent of the sequent will decompose to a positive signed atom in the proof net, while the neighboring atom of the same label will show the opposite sign (cf. Example 1). With the parentheses in the right place, an axiom link connecting the two atoms will not cross a boundary determined by them; with parentheses in the wrong places, the positive atom will be contained within a boundary which does not contain its counterpart negative (cf. Example 2). This argument extends by a structural induction to more complicated formulae. In essence, the device of the boundaries is a way of checking the grouping action of the parentheses in the proof net.

Example 4 shows that care must be taken to draw the parenthetical boundaries when subformulae involving "useless types" occur in the sequent. Observe that the subformula C/D has types C and D which are "useless", in that they do no work in reducing the sequent. When this is the case, we must draw the boundary all the way around the axiom links which connect the decompositions of the occurrences of C/D. After drawing the boundaries appropriately, we observe that once again no axiom link coming from a positive atom crosses a boundary which contains it. The proof structure for this provable sequent is then a proof net, under our newly formulated condition.

Example 4



We have at last descended all the way down the substructural hierarchy. As the logics became more stringent in dealing with a specific arrangement of formulae, the conditions on proof nets became accordingly more geometrical. Moreover, we noted that for those logics that do not deal with a specific arrangement of the formulae, it was possible to invoke the extremely compressed proof format of the matrix method. It is the author's hope that this unified discussion has illuminated the ways in which the "geometry" required of formulae in a logical consequence relation ends up being encoded into the "geometry of proofs" validating sequents for the logic. There are probably also some connections that could now be made with work that has explicitly represented proof nets topologically as cell complexes (e.g., Métayer 2001).

6. Complexity issues

While complexity of proof methods is not our focus here it must at least be mentioned, since the improvement in efficiency offered by compressed proof objects is a major reason for their promotion and study. The complexity of the decision problem in MLL has been shown to be NP-complete (Kanovich 1991), so no theorem-proving scheme can ever really be tractable. The best that can be hoped for is minimal intractability. Proof nets have mostly been studied for the time complexity of the proof verification problem, and along these lines, the Danos contraction condition as described above has complexity $O(n^2)$ in the size of the proof structure Moot (2002). Guerrini (1999), however, showed how to convert the contraction algorithm into one with linear complexity. Another way of developing a linear time correctness check was shown by Murawski and Ong (2000). Given the existence of linear-time algorithms to check correctness of a proof structure, the origin of the overall NP complexity is therefore the proof *construction* due to the sheer number of possible proof structures to be checked, because how to create axiom links can be indeterminate after expanding the formula tree.

Turning to the matrix method, the way to check correctness of a matrix involves traversing all paths through it to see whether there exists a (linearly) spanning set of connections for it. Now, while a matrix appears to be a proof object of a truly minimal size and graphic intricacy, the worst-case complexity of checking a matrix would seem to be exponential, on the order of 2^n in the length of the formula. This can happen in the case of a formula that involves nested disjunctive subformulae, which will yield nested column matrices through which all paths must be traversed. For this reason, the matrix method was dismissed out of hand by Hughes (2006) as not even a "proof system", which has occasionally been defined (Cook & Reckhow 1979) as a system in which proofs can at least be verified, if not constructed, in polynomial time. The matrix method does bear the singular feature that actually constructing the proof object is deterministic and linear-time. But, to borrow a common adage, if logic were that easy everyone would be doing it. The matrix method's powerfully simple proof construction leaves a large debt to be paid on the other end of the deal, when the time comes to check it. So in rough terms, matrices are easy to build but potentially hard to verify (not unlike the case of truth tables),[4] while the opposite is true for proof nets. In practical applications, of course, all these considerations are less important than the

[4] Thanks to the referee for pointing out this similarity.

ultimate competition among the average-case complexities, and discussing that is beyond our scope here.

7. Identity of proofs

The *identity of proofs* problem remains a significant open research question in logic and proof theory (Došen 2003).[5] Simply put, for any given logical system this is the question of when two apparently different proofs (of the same formula) ought to be regarded as fundamentally "identical." While at first glance this issue appears tangential to the main track of the present paper, it has to be addressed because it has so often been a central concern in the community researching proof nets for the system MLL, among others. Proof nets have usually been promoted as addressing this question directly (Girard 1987), and have sometimes been claimed to actually solve the issue (Straßburger 2006). It will now be explained how such claims should be viewed as exaggerated.

There may in general be more than one proof net for a provable sequent, however there can often be fewer possible proof nets than possible sequent proofs, even in a Cut-free system. Each proof net has often been viewed as representing an equivalence class of sequent proofs modulo "spurious ambiguity", while distinct proof nets will sequentialize respectively to full sequent proofs which are "nontrivially" distinct (Straßburger 2006). In lecture notes (*op. cit.*), Straßburger goes so far as to claim a theorem stating that two sequent calculus proofs in MLL translate to the same proof net iff they can be transformed into each other using only "trivial rule permutations." Yet, such a theorem seems to be circular, for in order to have this result one must assert in advance precisely what kinds of sequent rule permutations are held to be trivial and which are nontrivial. But it is this last issue that remains fundamentally a matter of debate!

Moreover (as Došen pointed out to me), on Straßburger's analysis, two sequent proofs which differ only by the presence of a useless Cut rule must be held to be nontrivially distinct, because the one with Cut will translate to a proof net involving a Cut link. Yet there is broad agreement among logicians that a sequent proof involving Cut should be regarded as "identical" to its Cut-free variant. Proof nets, therefore, should not be seen to have solved the identity of proofs question for any logical system. As for the matrix of a sequent, there can be only one, so as a proof-theoretical

[5] This section owes a great debt to personal communications with Kosta Došen, and I herein communicate some of his arguments.

object it does not address the fundamental question of "identity of proofs" other than to trivialize it.

8. Concluding remarks

Past literature has rarely, if ever, connected all the topics and treatments touched on in the present paper. It is in the spirit of a new synthesis that the paper is offered, with the hope of a more complete understanding. We observe many connections between efforts to compress proof schemata, where the geometrical requirements of the compressed proof object arise out of the substructural nature of the logic. We also observe the connection between signed subformulae, which arise out of the concept of negation and the duality therefrom, and the "link" or "connection" notions which are central to the compressed proof objects, whether matrix or proof net.

Acknowledgements

Thanks are owed to Greg Restall for commenting on an early version, and to Kosta Došen for some important insights.

References

Abrusci, V. M. 1991. Phase semantics and sequent calculus for pure noncommutative classical linear propositional logic. Journal of Symbolic Logic 56. 1403–1451.

Abrusci, V. M. 1997. Lambek calculus, cyclic linear logic, noncommutative linear logic: Language and sequent calculus. In V. M. Abrusci and C. Casadio (eds.) Proofs and linguistic categories. Bologna: Cooperativa Libraria Universitaria Editrice Bologna. 21–48.

Andrews, P. B. 1981. Theorem proving via general matings. Journal of the ACM 28. 193–214.

Beth, E. W. 1959. The foundations of mathematics. Amsterdam: North-Holland.

Bibel, W. 1980. A theoretical basis for the systematic proof method. In P. Dembiński (ed.) Mathematical foundations of computer science 1980. Lecture Notes in Computer Science, vol. 88, 9th Symposium. Berlin: Springer. 154–167.

Cook, S. A. and R. A. Reckhow. 1979. The relative efficiency of propositional proof systems. Journal of Symbolic Logic 44. 36–50.

Danos, V. 1990. La logique linéaire appliquée à l'étude de divers processus de normalisation (principalement du λ-calcul). Doctoral dissertation. University of Paris VII.

Danos, V. and L. Regnier. 1989. The structure of multiplicatives. Archive for Mathematical Logic 28. 181–203.

Došen, K. 2003. Identity of proofs based on normalization and generality. Bulletin of Symbolic Logic 4. 477–503.

Dyckhoff, R. and L. Pinto. 1999. Permutability of proofs in intuitionistic sequent calculi. Theoretical Computer Science 212. 141–155.

Fulop, S. A. 2004. On the logic and learning of language. Victoria: Trafford.

Gabbay, D. and N. Olivetti. 2000. Goal-directed proof theory. Dordrecht: Kluwer.

Galmiche, D. 2000. Connection methods in linear logic and proof nets construction. Theoretical Computer Science 232. 231–272.

Gentzen, G. 1934. Untersuchungen über das logische Schliessen. Math. Zeitschrift 39. 176–210, 405–431.

Girard, J.-Y. 1987. Linear logic. Theoretical Computer Science 50. 1–102.

Groote, P., de and F. Lamarche. 2002. Classical non-associative Lambek calculus. Studia Logica 3. 355–388.

Gross, J. L. and T. W. Tucker. 1987. Topological graph theory. New York: Wiley.

Guerrini, S. 1999. Correctness of multiplicative proof nets is linear. In 14th IEEE symposium on logic in computer science. 454–463.

Hughes, D. J. D. 2006. Proofs without syntax. The Annals of Mathematics 164. 1065–1076.

Kanovich, M. 1991. The multiplicative fragment of linear logic is NP-complete. Ms. University of Amsterdam.

Kelly, G. M. and S. Mac Lane. 1971. Coherence in closed categories. J. Pure and Applied Algebra 1. 97–140, 219.

Lamarche, F. and C. Retoré. 1996. Proof nets for the Lambek calculus – an overview. In Proceedings of the 3rd Roma workshop: Proofs and linguistic categories. 241–262.

Lambek, J. 1958. The mathematics of sentence structure. American Mathematical Monthly 65. 154–170.

Lambek, J. 1961. Structure of language and its mathematical aspects. In R. Jakobson (ed.) Proceedings of Symposia in Applied Mathematics. Providence, RI: American Mathematical Society. 166–178.

Métayer, F. 2001. Implicit exchange in multiplicative proofnets. Mathematical Structures in Computer Science 11. 261–272.

Moortgat, M. 1997. Categorial type logics. In J. van Benthem and A. ter Meulen (eds.) Handbook of logic and language. Amsterdam & Cambridge, MA: Elsevier & MIT Press. 93–178.

Moot, R. 2002. Proof nets for linguistic analysis. Doctoral dissertation. Utrecht University.

Morrill, G. V. 1994. Type logical grammar: Categorial logic of signs. Dordrecht: Kluwer.

Murawski, A. S. and C.-H. L. Ong. 2000. Dominator trees and fast verification of proof nets. In 15th annual IEEE symposium on logic in computer science. 181–191.

Pentus, M. 1997. Equivalence of multiplicative fragments of cyclic linear logic and noncommutative linear logic. In S. Adian and A. Nerode (eds.) Logical foundations of computer science. Berlin: Springer. 306–311.

Restall, G. 2000. An introduction to substructural logics. London: Routledge.

Robinson, E. 2003. Proof nets for classical logic. Journal of Logic and Computation 13. 777–797.

Roorda, D. 1991. Resource logics: Proof-theoretical investigations. Doctoral dissertation. Universiteit van Amsterdam.

Roorda, D. 1992. Proof nets for Lambek calculus. Journal of Logic and Computation 2. 211–231.

Smullyan, R. M. 1963. A unifying principle in quantification theory. Proc. Nat. Academy of Sciences 49. 828–832.

Smullyan, R. M. 1968. First order logic. Berlin: Springer.

Straßburger, L. 2006. Proof nets and the identity of proofs (Technical Report 6013). Rocquencourt: Inria.

Wallen, L. A. 1990. Automated proof search in non-classical logics: Efficient matrix proof methods for modal and intuitionistic logics. Cambridge, MA: MIT Press.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 219–232

Does literacy no longer need an institution to remain sustainable?

Some reflections on the impact of texting and messaging

MAIK GIBSON
SIL International
maik_gibson@sil.org

KEYWORDS

digital vitality
digital literacy
EGIDS
digital endangerment

ABSTRACT

We examine the impact of the recent phenomenon of digital writing, especially on minority languages that are thriving orally. It appears that successful literacy in minority languages formerly needed a strong institution, ideally supporting widespread education, in order for writing to be sustained. However, with web-based interaction, community norms and practices can spread and be sustained by informal interaction online and in text messaging. Examples from three language varieties (Rangi, Tunisian Arabic and Sheng) are given, where the main model for writing is not derived from a formal institution. This leads us to propose a modification of the interpretation of the EGIDS scale, such that EGIDS Level 5 may be a sustainable level for literacy rather than merely a step towards sustainability.

The future of many, or indeed most, of the world's languages is endangered; this is not a matter of debate. What is genuinely hard to predict is the trajectory of each language – which ones will be passed on to the next generation, and to the one after that. There is agreement on some general factors involved in language maintenance, shift and death: attitudes both internal and external; changes in lifestyle; contact and relationship with other groups; patterns of multilingualism etc. But how these factors play out in individual cases does not leave us with predictive power of what the precise future of each language is.

A related question is what the literacy status of a language is – some language varieties are used exclusively for oral communication, while others are used for writing, and for a variety of purposes. Just as there are social factors that influence the transmission of a language from one generation to another, so there are other factors which help the continued use of a language in a written form. The central question of this paper is whether the conditions which sustain literacy have changed with the advent of writing on digital platforms – more specifically, whether digital writing means that social institutions are no longer strictly necessary for the maintenance of community's use of the written form of the language.

On this topic of digital writing, Professor Kornai has written what has proved to be the seminal work: his 2013 paper *Digital language death*. In it he demonstrates that most languages have failed to ascend digitally, that is, to become vehicles of written digital expression, whether on a computer or phone. One of the key questions for linguists is whether digital writing presents an opportunity for minority languages to be used in new sociolinguistic domain (Fasold 1984, 60), or whether the digital environment is another factor in the decrease of prestige, and shift to another language. It seems clear that texting, messaging and the internet in general are something of both an opportunity and a danger to minority languages, depending on a wide variety of factors.

One of Kornai's key observations is that for digital language use to be vital, it must involve "active use in a broad variety of two-way contexts such as social networks, business/commerce, live literature, etc." (2013, 3). With this comment he rightly dismisses the presence of dictionaries and the like as sufficient conditions of digital vitality – interaction is the key, rather than the mere existence of the documentation, as important and laudable task as it is. We can build on this observation by recalling Abercrombie's (1963, 14) comment that "writing is a device developed for recording prose, not conversation" – digital writing is, on the other hand, often conversational, as it shares with conversation the in-the-moment interactivity that other forms of writing, fixed on paper, do not. As such, we can make the case that messaging has opened the door to genuinely conversational writing. And it seems to be the case that in writing, conversation makes the use of any vernacular more likely – whether an endangered, minoritised, or non-standard, variety of language.

Conversation is where most minority languages survive – in discussions at home, in informal gatherings, in the fields or bars. In multilingual societies one often finds languages of higher prestige used in education, formal writing and so on, but in general the vernacular or the non-standard is the appropriate variety to use when the communication is interactive. These conversations disappear from record the moment after utterance (interestingly enough, a practice shadowed by some messaging apps such as Snapchat). When used for in-the-moment person-to-person interaction, digital writing can mirror these spoken practices, and thus we see an impetus around the world to use non-standard or vernacular forms of writing, from textspeak (Crystal 2008) in English, to a multilingual situation where speakers write in a language normally reserved for speech. Digital writing is a more natural domain for the writing of otherwise rarely written languages. It seems reasonable to suggest that this use may then, at the very least, serve as a model for other writing, showing that writing the vernacular is a possibility.

We also note that the internet can increase exposure to other languages, and in some cases may be part of the hastening of a shift to a language which is perceived as having richer benefits (Karan 2011). But in this paper we have decided to focus on opportunities rather than the threats, which can look after themselves.

Just as it is easier to work on preventing a village from falling into a river rather than reconstructing it after it has fallen in ('a stitch in time saves nine'), so the most effective strategies of language preservation are those where the language is still vital, rather than cases where the language has already mainly been lost. The analogy applies also to digital engagement with minority languages – it is more likely to take root where the language has not yet started to fall into the river, that is, it is still used within all generations of the community in daily spoken communication.

In order to be able to address the question of whether digital media have introduced a new opportunity for communities to use their languages in written form, we introduce parts of the two frameworks that help raise the issue of what constitutes sustainable, vital use of language in written form, EGIDS and Kornai's measure of digital vitality.

EGIDS

In order to be able to talk of language endangerment and revitalisation with greater precision, we introduce a part of EGIDS (Lewis & Simons 2010; Simons & Lewis 2016) – an extension of Fishman's (1991) *Graded Intergenerational Disruption Scale*. Unlike the UNESCO framework of language vitality (Brenzinger et al. 2003) which views a language as vulnerable if it is not used in education, EGIDS' central question is whether "The language is used for face-to-face communication by all generations and the situation is sustainable" (Lewis & Simons 2016, 80). If this is the case, the language use is classed as *Vigorous*, and assigned at the very least a Level 6a (the lower the number, the higher the level of vitality). This is distinguished from Level 6b, *Threatened*, defined as "The language is used for face-to-face communication within all generations, but it is losing users" (*ibid.*, 81), which is the case when not all children within the language community are acquiring the communal language. For EGIDS the primary question for the future of the language is whether it is currently being transmitted orally to children, rather than questions of use in literacy or education, important as these can be. It is only once it is established that the language is at a minimum level of 6a, *Vigorous*, that other factors such as use of the language in writing are considered. Overall a language which does have written presence, but is losing speakers, is regarded as being less vital than one where oral transmission continues, but without literacy.

The next two levels, above 6a, do take into account the status of literacy in the language, and are:

- 5, *Developing*, "The language is in vigorous use, with literature in a standardized form being used by some, though this is not yet widespread or sustainable",
- 4, *Educational*, "The language is in vigorous use, with standardization and literature being sustained through a widespread system of institutionally supported education" (Simons & Lewis 2016, 80).

Level 4, *Educational,* shares with Level 6a the fact that it is a sustainable level – in contrast to 6b, where the loss of some speakers, will, without effective intervention, lead to a further degradation of the vitality status of the language. At Level 4, as long as an educational system is being sustained (with not just funding for the schools, but also some system for training teachers), the level is expected to be stable. In most cases the institution will have government backing, but other possibilities for sustainability exist, such as religious institutions, community organisations, or NGOs. In these cases, the level will be stable only as long as the institution sees the value of continuing the education in this language.

Level 5, *Developing*, is a natural stage on any transition from a language used purely for oral expression to one used in a sustainable educational system. There needs to be some development of literacy in a language used only orally before it can be used in education, such as developing a writing system, agreeing the vocabulary to be used, and developing materials. And a body of users (e.g., of those who will teach the language) must also develop. However, an insight of the EGIDS model is that this level, without a sustainable institution, is not stable. If there is activity at this level which does not continue, the language will naturally slip back to level 6a; even to sustain this level needs constant activism. This is an important observation, which can guard against the false optimism of developing a writing system which a few people use, under the assumption that these practices will spread themselves. In most cases, unless there is a champion within the community who then is key in establishing an institution, literacy will not be more widely adopted.

A diagram showing the relevant part of the "language mountain" is shown below. The full diagram of the whole EGIDS scale is shown in Simons & Lewis (2016, 116). A flat area represents an EGIDS level which is sustainable. In the diagram we see that this is the case for levels 4 and 6a. Level 6b is a situation which is inherently unstable – without communal intervention, the language is likely to move eventually to a level where the youngest generation of speakers has reached adulthood (Level 7). Moving up to 6a would require substantial effort, signified by the steep gradient. In the same way, Level 5 is seen as a level where without institutional support in teaching the language, the language will most likely revert to use only in oral domains. We argue here in this paper that the gradient is perhaps not so steep at Level 5, if a community starts writing its language in digital media.

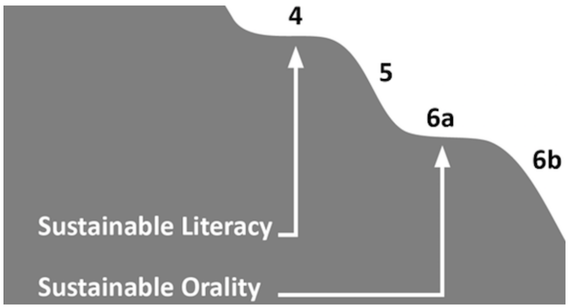


Figure 1: Levels 4–6 in the EGIDS, adapted from Simons & Lewis (2016, 116)[1]

Measuring digital vitality

EGIDS was not designed with digital writing in mind, though in their later work, Lewis and Simons (2016, 195) do mention that "mobile telephones, text messaging, and the worldwide web, can also be identified as venues for local-language use and sharing of identity-re-enforcing knowledge". Kornai

[1] Used by permission, © SIL International (Simons & Lewis 2016), further redistribution prohibited without permission.

(2013) is to some extent an attempt to apply the principles behind EGIDS to digital writing, backed up with empirical evidence of the digital presence of different languages obtained through web crawling. The resulting scale is much simpler than EGIDS, of which we have presented only a part.

Kornai presents four categories of digital vitality. I refer the reader to his work for a fuller account.

Thriving	**T**
Vital	**V**
Historical/Heritage	**H**
Still	**S**

Thriving and *Still* are somewhat self-explanatory; the key distinction for our purposes in this paper is the difference between *Vital* and *Heritage*. In both cases the language is present on the internet in some form, but with the *Vital* category it is used between speakers for two-way communication, as opposed to *Heritage* (or *Historical*), in which case the "language is not used by native speakers (L1) for communication in the digital world" (2013, 2). At the *Heritage* level there may be documentation, or even Wikipedia entries, or a dictionary, but the two-way interaction, also key for Level 6a *Vigorous* in EGIDS, is absent. Further categories of digital vitality, *Emergent* and *Latent*, are proposed by Gibson (2015; 2016), primarily to help identify which activities may be useful to enable digital ascent, in much the same way as the *Sustainable Use Model* (Lewis & Simons 2016) is focused on helping communities identify useful activities for securing the future of their language. EGIDS, which it builds upon, is primarily a tool of description and analysis.

It is evident that if a language is classified as being *Vital* in digital vitality, then it will be normally at the very least at EGIDS Level 4, i.e., supported by a sustainable institution. In fact, for the *Vital* category, Kornai (2013:9) states "This group contains about two thirds (66%) of the EGIDS 1 languages; less than half (46%) of EGIDS 2; 13% of EGIDS 3; 8% of EGIDS 4; 2% of EGIDS 5; and less than 1% of all higher classes." We see that in fact most languages at the lower end of sustainable literacy are not judged to be digitally vital – some may well be borderline between vital and still. But the presence of even some languages below EGIDS Level 4 at the digital *Vital* level demonstrates that these language communities have found ways to use the language in written communication with each other, without a sustainable institution. In cases where this use is marginal (e.g., only a small portion of the community uses the language in written communication), and EGIDS level 5 would seem appropriate. EGIDS figures listed in Simons & Fennig (2017) are not the final word, and are subject to change in the presence of new data – that Kornai has found digital presence of a language at the *Vital* level is surely sufficient to recategorise these languages as at 5 at the very least, with the possible exceptions of where the digital usage is not matched by oral communal usage.

Some examples of digital writing

We now report on three examples of digital writing in language varieties which do not have any significant institutional support. In each case we give but a brief example, and discuss the context in which this use is found. The examples give but a small flavour of this kind of interaction, and are here to demonstrate that digital writing can exist where there is no institution which teaches people how to read and write the language variety, or decrees which is the correct way of writing it. Rather than giving detailed documentation of multiple examples, or rates of use, we demonstrate usage, that is, however, not isolated, but reflective of broader trends.

For people to be able to write to each other effectively, there do need to be some conventions – an agreement on which script or scripts are appropriate, for example, and which letters are appropriate to represent which sounds. Total agreement on these issues is not necessary – just as in oral communication, we can effectively communicate where there is a difference in dialect, but there are also limits to effective communication where divergence increases. In each of the cases the language users have learnt to read and write in other language varieties, and have transferred this skill to a variety not taught at school. However, the development of orthographic conventions – normally the role of an institution, is arrived at by some level of negotiation, whether overt or developmental.

Under Kornai's paradigm these three cases can be described as *Vital*, with evidence of two-way communication going on between community members. In two of the three cases there is a question over whether the target variety is in fact an independent language, but in some ways this is also a moot question – we will see that conventions for writing a language variety have come into play without significant intervention from an institution – the community of web users can be said to have taken over this function more often performed by an institution. The definition of what constitutes separate languages is notoriously difficult and controversial (e.g., see Simons & Fennig 2017 on *The problem of language iden-*

tification),[2] so we have chosen to remain non-dogmatic here – the issue is broadly similar, whether the variety is defined as a separate language, or as a non-standard variety of another language. We will start by looking at the case which is indisputably a separate language, Rangi of Tanzania.

Rangi (ISO 639-3: LAG) is listed in Simons & Fennig (2017) as at 6a on EGIDS, and is spoken by over 400,000 people, but not used in the education system. There are some Biblical and other texts – the Bible uses barred vowels in addition to the typical Latin five vowels, the result of careful phonological analysis being applied to an orthography. There is however no agreed standard orthography, but the most available model is that of the Bible. As for all of Tanzania, the media of instruction in school are Swahili and English. However, there is a Facebook page,[3] which is for community members to use the language. The page does not function exclusively in Rangi, with Swahili also present, in line with the multilingual practices of the community. It is important to note that the page is focused on the use of the language by the community, and as such is less of strong example of vital digital use than the other two cases below, where the name of the language variety is not on the page – the interaction in those cases has no tinge of language activism –, the message is clearly of more importance than the medium. In the case of Rangi, unlike the two that follow, we are unable to confirm that what is observed on a Facebook page is reflective of broader communal practices commonly found elsewhere.

I will quote a couple of texts on the site, merely to demonstrate usage related to the barred vowels of the Biblical text. The first example (dated 5 December 2017), contains the following text: "Kei sɨ kwa mɨrɨmo yaanyu tʉkʉ, sa mʉʉntʉ yoyoosi adɨɨre kwiivaa kɨpeembe". The reader will note the use of barred vowels – this is a Bible verse, which is then repeated in both Swahili and English. However other posts do not demonstrate these barred vowels (e.g., a post from 30 December 2017 has "Kalarira saka eeeh mukulu mikate yosi jei na ndii wuuuu kibirya mpia sana"). We can assume that the Bible passage was copied and pasted from a soft version of the Bible. However, other online use has eschewed the symbols that are not used in Swahili (the barred vowels), and which are also more of a challenge to find on a phone or computer keyboard. This is a tendency which seems quite common when community members write minoritised languages – making do with simpler solutions, and not necessarily making an effort to reflect a standard form of the language. This demonstrates that even

[2] https://www.ethnologue.com/about/problem-language-identification

[3] https://www.facebook.com/groups/TuluusikeKilaangi/

where there is a model (in this case the Bible) where phonemes are orthographically distinguished, speakers do not always adopt these solutions, often adopting solutions which are easier to implement, or conform more closely to orthographic models already available. In this case, both factors might be at play in the lack of uptake of the barred vowels. Overall, the language may have reached the *Vital* (or at the very least a Borderline) level in the digital sphere, through the presence of the Facebook group, and this should also qualify the language to be deemed at Level 5 on EGIDS, and that this level may be sustainable. How much deliberate effort is being made by those promoting the written use of the language is not clear. There is less evidence of more widespread use of the language than in the following two examples. The conventions followed seem to be as close a match to Swahili as possible.

Tunisian Arabic (ISO 639-3: AEB, also listed as part of the Arabic macrolanguage ARA), is spoken by over 11 million people (Simons & Fennig 2017). Formal writing in Tunisia is generally either in Standard Arabic or French, and there is no official standard for Tunisian Arabic. However many advertising slogans are written in the variety, and there has been an increased use of the oral variety in broadcast media over the last twenty years. In general the spoken norms of the capital Tunis are followed in such writing. Children are not taught to read and write in this variety, despite its being the home variety of over 99% of the population. It is listed at level 3 on EGIDS, as it functions as the vehicular language outside formal domains, even if without any formal status, or a standardised variety.

I will present just one piece of data which is demonstrative of how the variety is used in writing: "N3ichou elwahm w msad9inou". This is found on the public page of a Tunisian Facebook user, written by another user in response to a post. Such usage is extremely common on Facebook, not restricted to pages which mention Tunisia or Tunisian Arabic. Here we find Tunisian Arabic written in an adapted form of Latin script, with numbers added to this to represent sounds for which there is no good Latin equivalent. In this case "3" stands for the voiced pharyngeal fricative [ʕ], which in Arabic is written ع, which looks like a laterally inverted 3. Likewise "9" represents a voiceless uvular plosive /q/, which is written with the Arabic letter ق. Again, the reader can see the similarity. In Tunisia, normal Latin transcription of Arabic names (for example on road signs) uses "k" for this sound, which is ambiguous as it also represents /k/. This system, which can also use 2, 5 and 7 as letters, is sometimes known as Arabizi (e.g., Darwish 2013). This comment, which means "We are living the illusion and believing it", is followed by a comment from another user, written in Standard Arabic, and in Arabic script. Many other comments are written in Tunisian Arabic, but in Arabic script. The conventions for writing Arabizi (or the Arabic chat alphabet) are not taught in school, but learnt through interaction with others online. The form represented is also clearly non-standard, representing Tunisian speech.

We are not claiming that Tunisian Arabic is a separate language from Standard Arabic, as that is not how most Tunisians view their language practices. The innovative choices that are made draw on general education, including learning to read and write in French, but do not reflect any practice taught at any formal institution: specifically writing the Tunisian form of Arabic; writing it in Latin characters; and using numbers to represent certain sounds. These conventions are however shared across the community of Tunisian web users. While certain conventions exist, we cannot talk of an enforced standard, but rather a few parameters which are shared and aid effective written digital communication. A more detailed coverage of this phenomenon would show a great amount of variation in written forms – we cannot say that there is a standardised variety of written Tunisian Arabic, but there are nevertheless some shared practices which have been adopted by a wide community. So we can say here that the institutions of Tunisia are what have taught people to read and write – in Arabic and French, but that there are some orthographic conventions which have developed (not just in Tunisia, but throughout the Arab World) through communal use. These conventions are normally promoted by an institution, but here the institution, if any, is that of the body of the internet users. This reflects spoken norms, in that what is deemed appropriate for many spoken language varieties is not decided by an institution, but by an aggregate of the speakers themselves. An example is that the Tunis variety of Tunisian Arabic functions as a *de facto* prestige variety of Tunisian Arabic, without any institutional support (Gibson 2002).

Our final example is *Sheng*, an urban variety of Swahili spoken in Nairobi featuring much lexical innovation and language mixing, or translanguaging. Some scholars (e.g., Bosire 2006; Rudd 2008) in fact claim that it is an independent language – in any case, this question of the status of Sheng does not impact our judgement as to whether users of digital media are able to establish some level of orthographic conventions among themselves without intervention by an official body. Sheng does not have an ISO 693-3 code, as it has not met the criteria for inclusion as a separate language from Swahili. Sheng usage in written communication is also found widely on platforms such as Facebook, and has a lot of use in written advertising slogans, a trait it shares with Tunisian Arabic. In such cases it is sometimes referred to as *Kenyanese* (Erastus 2013), to avoid the sometime negative associations of the term *Sheng.*

The example given here is perhaps not of quite the same nature as those for Rangi and Tunisian Arabic, as it is taken from the Facebook page of "DJ Boyie", a fictional character who is part of the Shujaaz.FM[4] multimedia platform. An example sentence is *Nadai ku'get the best hustlas ii mwaka*, which demonstrates some of the features of Sheng, in that it uses innovative lexis (*dai* for 'to want') and English expressions (in this case a whole noun phrase). Note the spelling of *hustlas*, presumably influenced by African American norms also found in the *z* of the *Shujaaz* 'heroes' in the platform's name. We also have the English root 'get' used with a Swahili infinitive marker, with the convention of using an apostrophe to separate the English root from the Swahili affixation. And then we have *ii mwaka* 'this year', where *ii* is the Swahili *hii* 'this' with the *h* dropped, and the standard Swahili order of the modifier and head inverted. Shujaaz.FM has also used hyphens (such is in *ku-come* 'to come') to separate Swahili affixes from English roots – in this case, the hyphen helps with recognising that the pronunciation of the root should follow English and not Swahili pronunciation rules. We also find no marking from DJ Boyie *nlibuy* 'I bought', along with a contributor using *imeneglect* 'It has neglected'. However users of the website do not seem to have adopted either the convention of the hyphen or apostrophe. Instead strategies include leaving a space between the Swahili morphology and English root *wana stay* 'they stay', *nisha cheki* 'I already checked'.

While we might consider that Shujaaz.FM is some sort of institution, even its conventions do not yet seem to have been adopted. But written communication proceeds nonetheless. Again we see that the educational institutions of Kenya have effectively introduced literacy in Swahili and English, and this knowledge has been used in writing this non-standard, mixed, variety. But when confronted with cases which are somewhat alien to either language – using an English verbal root with Swahili verbal morphology, users resort to a variety of strategies, the dominant ones seeming to be either splitting the morphology from the root or combining them with no marking. The solution primarily followed by *Shujaaz.FM*, using an apostrophe (or a hyphen) to mark the boundary between forms originating in different languages, despite its rationale, does not seem to have been adopted by the broader community. Thus, as with Rangi and Tunisian

[4] https://www.facebook.com/DJBoyie/

Arabic, the usage of the broader community has not been fixed by an institution, but is negotiated among the community, thus circumventing one of the functions of the institution.

Is digital writing creating a new kind of sustainability for literacy?

Kornai's paradigm of digital endangerment and death has driven us to ask questions which bring us back to evaluate more closely EGIDS, and the effect that digital writing has on the broader context of the use of a language variety in writing. In each of the cases we have seen, there is no institutional support for the conventions that speakers appear to be using. As previously noted, these conventions are loose, with much room for latitude, but the very fact that people continue to use these forms of writing shows that effective communication is taking place – otherwise it would not be sustained. The case for this claim is the strongest in the case of Tunisian Arabic – using Arabizi is a widespread phenomenon, seen even in advertising in Lebanon. Interestingly, all three cases are to some extent a challenge to Kornai's (2013, 4) statement that "languages without mature writing systems are unlikely to digitally ascend". In each case we note that the writing system is not mature, as it is not standardised, and conventions are still being negotiated, akin to the negotiation and valorisation of social norms for speech. However, it would seem sensible to concede that the challenges for digital ascent of non-standardised varieties might well be greater than those of languages with an accepted standard version, due to extended use.

Now, in no case am I arguing that there is no institutional support for writing – we can safely assume that all the people using these varieties have had an education, but one which taught them how to write another language or languages. Conventions for those languages were learned, sometimes fully, sometimes imperfectly perhaps, and some of those conventions have been drawn into their literacy practices when writing other varieties. There is obviously a feeling that at least one of the scripts that has been learnt is an appropriate vehicle for writing something other, at least on social media. While Rangi exhibits some greater phonological complexity than Swahili, with a seven- rather than five-vowel system, as well as distinctive tone, this has not been an insurmountable barrier to effective writing in a system more closely aligned to Swahili phonology. However, where tone bears a heavier load, and the language of education does not use it, we can imagine greater challenges in successful community-led orthography construction. Even though Tunisians often use Latin script (alongside Arabic script) to write their vernacular, the system also references Arabic script in a way that formal transcription standards do not – in fact their use shows the deficiencies of the Latin transcription system used for signage in Tunisia.

We note that the formal education system has its role to play in the implementation of vernacular writing, probably along with some level of linguistic similarity between what is learnt at school and what is spoken at home or on the street (and all our cases have been at the less challenging end of the spectrum in this respect). We suggest that at the very least, where these conditions have been met, literacy may take hold without the involvement of a sustainable institution. Again, we have demonstrated this only for limited domains, with limited examples, and have not yet demonstrated that this use effectively spreads to more formal domains – there is much room for further study. But we would like to ask the question, in the environment of widespread digital literacy, whether incipient use of a language for writing might now in some cases be sustainable *without* an institution behind it. And we ask the question with an idea of the answer – that, yes, there is evidence of digital writing practices establishing themselves without systematic institutional support. We might expect the use of the vernacular in writing to be even more widespread in private messaging, including texting, but we have more of a challenge in observing this, as opposed to the semi-public domain of Facebook which have been used for this paper. Therefore Level 5 in EGIDS should no longer be so hard to climb up in the language mountain as in the picture shown above.

Having addressed a question of theory, we are driven to ask what difference this might make in practice. If, as seems to be the case, digital spaces are places where orthographic innovations can take root, then there are implications for those of us concerned with applying our linguistic knowledge for the benefit of minority ethnolinguistic communities. We may wish to work with them in identifying which activities are the most likely to be effective in helping them meet their own goals for the uses of their language. From the evidence presented here, it would seem that messaging using mobile phones will be a fruitful, or perhaps essential, step for languages which are at EGIDS levels 6a or 5 – more considerations for effective engagement are suggested in Gibson (2015; 2016). Digital literacy seems to present an opportunity for further written development of varieties which employ it; how effective this may be is yet to be seen.

References

Abercrombie, D. 1963. Problems and principles in language study. London: Longman.

Bosire, M. 2006. Hybrid languages: The case of Sheng. In O. F. Arasanyin and M. A. Pemberton (eds.) Selected proceedings of the 36th Annual Conference on African Linguistics. Somerville, MA: Cascadilla Proceedings Project. 185–193.

Brenzinger, M., A. Yamamoto, N. Aikawa, D. Koundiouba, A. Minasyan, A. Dwyer, C. Grinevald, M. Krauss, O. Miyaoka, O. Sakiyama, R. Smeets and O. Zepeda. 2003. Language vitality and endangerment. Paris: UNESCO Ad Hoc Expert Group Meeting on Endangered Languages.

Crystal, D. 2008. Txtng: The gr8 db8. Oxford: Oxford University Press.

Darwish, K. 2013. Arabizi detection and conversion to Arabic. http://arxiv.org/pdf/1306.6755.pdf

Erastus, K. F. 2013. Manipulation of Sheng in advertisements and awareness campaigns in selected businesses in Kenya. Paper presented at the African Urban and Youth Language Conference, Cape Town, July 2013.

Fasold, R. 1984. The sociolinguistics of society. Oxford & New York: Blackwell.

Fishman, J. A. 1991. Reversing language shift. Clevedon: Multilingual Matters.

Gibson, M. 2002. Levelling in Tunisian Arabic: Towards a new standard? In A. Rouchdy (ed.) Language contact and language conflict in Arabic. Variations on a sociolinguistic theme. London & New York: Routledge. 24–40.

Gibson, M. 2015. A framework for measuring the presence of minority languages in cyberspace. In E. Kuzmin, S. Bakeykin, T. Murovana, A. Parshakova and N. Zaikova (eds.) Linguistic and cultural diversity in cyberspace. proceedings of the 3rd International Conference (Yakutsk, Russian Federation, 30 June–3 July, 2014). Moscow: Interregional Library Cooperation Centre. 61–70.

Gibson, M. 2016. Assessing digital vitality: Analytical and activist approaches. In C. Soria, L. Pretorius, T. Declerck, J. Mariani, K. Scannell and E. Wandl-Vogt (eds.) CCURL 2016. Collaboration and Computing for Under-Resourced Languages: Towards an Alliance for Digital Language Diversity. Proceedings. European Language Resources Association (ELRA). 46–51.

Karan, M. E. 2011. Understanding and forecasting ethnolinguistic vitality. Journal of Multilingual and Multicultural Development 32. 137–149.

Kornai, A. 2013. Digital language death. PloS ONE 8. DOI 10.1371/journal.pone.0077056.

Lewis, M. P. and G. F. Simons. 2010. Assessing endangerment: Expanding Fishman's GIDS. Revue roumaine de linguistique 2. 103–119.

Rudd, P. 2008. Sheng: The mixed language of Nairobi. Doctoral dissertation. Ball State University.

Simons, G. and M. P. Lewis. 2016. Sustaining language use: Perspectives on community-based language development. Dallas, TX: SIL International.

Simons, G. F. and C. D. Fennig (eds.). 2017. Ethnologue: Languages of the world. 20th edition. Dallas: SIL International. Online version: http://www.ethnologue.com/.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 233–245

Skeleton in the Euclidean closet

ZALÁN GYENIS
Jagiellonian University and Eötvös Loránd University
zalan.gyenis@gmail.com

KEYWORDS

Euclidean automata
skeleta

ABSTRACT

Euclidean Automata have been introduced in Kornai (2014a) to model a phenomenon known as "being in conflicted states". This brief note gives a further look on Euclidean Automata and takes the first steps in studying skeleta and representability and the logical characterization of languages accepted by Euclidean Automata.

1. Introduction

Euclidean Automata (EA) has been introduced, motivated and further studied in Kornai (2014a) and Kornai (2014b). EA are slight generalizations of the classical finite state automata: EA can take continuous parameters as input and are used in Kornai (2014a) to analyze the situation of being in a conflicted state. Intuitively, being in a conflicted state is modeled by an EA not as a single state (of the EA) but rather as a set of "nondeterministic" states that are represented as overlapping parts of the input parameter space. Let me recall the precise definition of Euclidean Automata.

Definition 1.1 (Kornai 2014b)**.** A Euclidean automaton (EA) over a parameter space Σ is defined as a 4-tuple (Q, I, F, α) where $Q \subseteq 2^{\Sigma}$ is a finite set of states given as subsets of Σ; $I \subseteq Q$ is the set of initial states; $F \subseteq Q$ is the set of accepting states; and $\alpha : \Sigma \times Q \to Q$ is the transition function that assigns for each parameter setting $v \in \Sigma$ and each state $q \in Q$ a next state $\alpha(v, q)$ that satisfies $v \in \alpha(v, q)$. □

In Kornai (2014a) the EA is called deterministic if $q \cap s = \emptyset$ for different $q, s \in Q$ and complete if $\cup_{q \in Q} q = \Sigma$. Throughout we will work with complete EA's only, the reason is that for $v \in \Sigma - \cup_{q \in Q} q$ the condition

$v \in \alpha(v, q)$ does not make sense, hence either one keeps α to be undefined on certain input parameters v or switches to an equivalent EA with parameter space $\cup_{q \in Q} q$. For simplicity we assume throughout that the set of initial states I contains a unique state which we denote by `start`. If one permits several initial states he needs to complicate the results accordingly. In applications drawn in Kornai (2014a;b) the alphabet Σ consists of vectors from a continuous parameter space, typically $\mathbb{R}^n$, however it also makes sense to consider the definition of an EA when Σ is a finite set, especially if one considers skeleta of EA's, as we do in section 2.

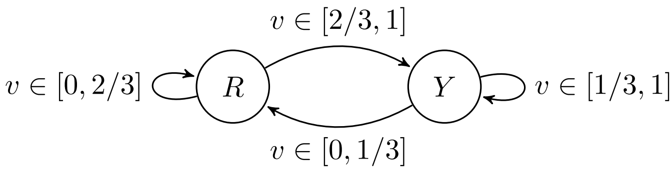
A typical application in Kornai (2014a) is the heap (Sorites paradox) presented in the Sainsbury & Williamson (1997) manner: Consider the line segment $[0, 1]$ colored so that the left-hand region is red, and there is a very fine, continuous, gradual change of shades reaching the right-hand side region that is colored yellow. The line is covered by a tiny window that exposes only a small region. We move the window very slowly starting from the left-hand side towards right and after each move one is asked about the color of the segment exposed by the window. But the window is so small relative to the line segment that in no position can you tell the difference in color between what you can see at the two sides of the window. It seems that you must call every region red after every move, and thus you find yourself in the paradoxical situation calling a yellow region red.

2. Skeleta and representability

Kornai modeled the heap paradox by an EA in a similar manner as we do below (to make life easier we give a somewhat simplified model, but the differences are inessential for the sake of the example). Let's say $[0, \frac{1}{3}]$ is "clearly red", $[\frac{2}{3}, 1]$ is "clearly yellow" and $[\frac{1}{3}, \frac{2}{3}]$ is this "hard to tell, orange" range. Our EA will have 2 states: red (R) and yellow (Y) respectively with $R = [0, \frac{2}{3}]$ and $Y = [\frac{1}{3}, 1]$. Note that the two states overlap exactly in the "problematic" region. Starting from the red state the machine gets input from the continuous parameter space $\Sigma = [0, 1]$. The machine is defined as follows:

$$\alpha(v, R) = \begin{cases} R & \text{if } v \in [0, \frac{2}{3}] \\ Y & \text{otherwise.} \end{cases}, \qquad \alpha(v, Y) = \begin{cases} Y & \text{if } v \in [\frac{1}{3}, 1] \\ R & \text{otherwise.} \end{cases}$$

In figure:



In the entire fuzzy orangish region $[\frac{1}{3}, \frac{2}{3}]$ the model shows hysteresis: if it came from the red side it will output red, if it came from the yellow side it will output yellow. To get a better understanding of how EA works, Kornai hints at skeletonizing EA's. The skeleton of an EA is defined in Kornai (2014b) as follows.

Definition 2.1 (Kornai 2014b)**.** The skeleton of an EA is a standard FSA whose alphabet corresponds to canonical representatives from each Boolean atom of Q. □

In the deterministic case (where all the states of the EA are disjoint) there is a correspondence between input letters and automaton states. However, in the nondeterministic case (where states are not necessarily disjoint) we may not be able to select distinct canonical representatives for each state (or for the Boolean atoms). In this case skeleta should be understood as "subjective EA" (cf. Kornai 2014b). The definition seems a bit vague as it is not completely clear how to chose the so called canonical representatives (or the subjective representatives), moreover, Q may have no Boolean atoms (cf. Example 2.4 below). A key for the clarification is the observation that some inputs are totally indistinguishable no matter what state the machine is in. To obtain a definition for the general case, fix an EA $\alpha : \Sigma \times Q \to Q$ and for $v, w \in \Sigma$ write

$$v \sim w \iff (\forall q \in Q)\ \alpha(v, q) = \alpha(w, q) \tag{1}$$

Then $\sim$ is an equivalence relation on Σ. Moreover it is a congruence of α as for any input sequences $\langle v_1, \ldots, v_n\rangle$ and $\langle w_1, \ldots, w_n\rangle$, $v_i \sim w_i$ implies

$$\begin{aligned}
\alpha(v_1, \ldots, v_n, \texttt{start}) &= \alpha(v_n, \alpha(v_{n-1}(\cdots, \alpha(v_1, \texttt{start})))) &(2)\\
&= \alpha(v_n, \alpha(v_{n-1}(\cdots, \alpha(w_1, \texttt{start})))) &(3)\\
&= \cdots &(4)\\
&= \alpha(w_1, \ldots, w_n, \texttt{start}) &(5)
\end{aligned}$$

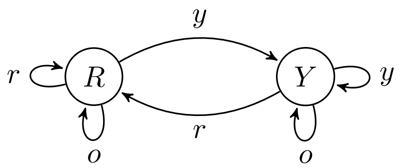
After this preparation we can redefine the concept of skeleta as follows.

Definition 2.2. The skeleton of the EA $\alpha : \Sigma \times Q \to \Sigma$ is the standard FSA $\bar{\alpha} : \Sigma/_\sim \times Q \to Q$ defined by the equation

$$\bar{\alpha}(v/_\sim, q) = \alpha(v, q)$$

Since $\sim$ is a congruence, $\bar{\alpha}$ is well-defined. □

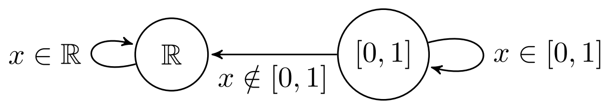
If we apply Definition 2.2 to the heap example given above we end up in the finite state machine figured below, which, unsurprisingly, is exactly the FSA sketched in Kornai (2014a). (Here the input letters r, y and o stand for red, yellow and orange, respectively)



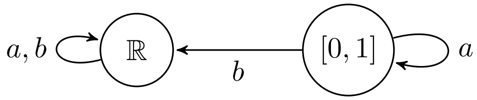
Observe that $\Sigma/_\sim$ is always finite. This is because the original EA has finitely many states only, hence we can have a finite number of possibilities not to fulfill $\alpha(v, q) = \alpha(w, q)$ for input letters v, w. This results in a finite number of equivalence classes of $\sim$. Unfortunately, $\bar{\alpha}$ is no longer an EA as $Q \not\subseteq 2^{\Sigma/\sim}$. It would be handy to define the skeleton of an EA as another EA over the finite alphabet $\Sigma/_\sim$ by letting $Q/_\sim = \{q/_\sim : q \in Q\}$ where $q/_\sim = \{v/_\sim : v \in q\}$. However, the automaton $\beta : \Sigma/_\sim \times Q/_\sim \to Q/_\sim$ defined in the obvious manner $\beta(v/_\sim, q/_\sim) = \alpha(v, q)/_\sim$ is not always well defined as the next examples show.

Example 2.3. Below we give an example for an EA the skeleton of which can be represented as an EA. Let the alphabet (parameter space) be $\Sigma = \mathbb{R}$ and the set of states is $Q = \{\mathbb{R}, [0, 1]\}$. Let α be the EA figured below defined by the equations

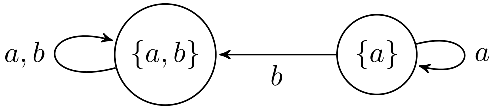
$$\alpha(x, \mathbb{R}) = \mathbb{R}, \qquad \alpha(x, [0, 1]) = \begin{cases} [0, 1] & \text{if } x \in [0, 1] \\ \mathbb{R} & \text{otherwise.} \end{cases}$$



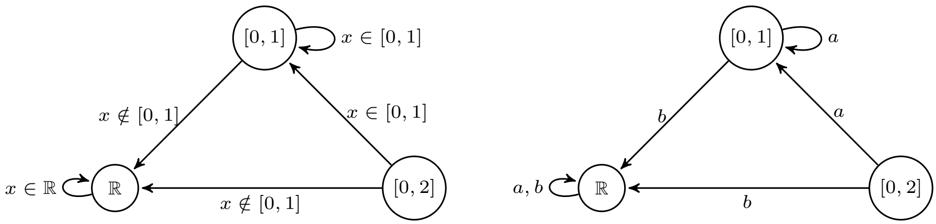
The equivalence relation $\sim$ will have two classes: $\Sigma/_\sim = \{[0, 1], \mathbb{R} - [0, 1]\} = \{a, b\}$ and the skeleton $\bar{\alpha}$ looks like



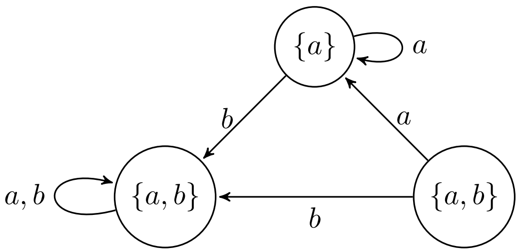
Since $\mathbb{R}/_\sim = \{a, b\}$ and $[0,1]/_\sim = \{a\}$, the skeleton is a EA over $Q/_\sim = \{a, b\}$:



Example 2.4. A small modification on Example 2.3 prevents the skeleton to be represented by an EA. Here Σ is as before but $Q = \{\mathbb{R}, [0,1], [0,2]\}$. The EA α is as figured below on the left-hand side.



The equivalence relation $\sim$ has two classes again: $\Sigma/_\sim = \{[0,1], \mathbb{R} - [0,1]\} = \{a, b\}$ (note that elements of $[0,2] - [0,1]$ behave exactly the same way elements of $\mathbb{R} - [0,1]$ do). Thus the skeleton can be figured as above on the right-hand side. Note, however, that $Q/_\sim = \{\{a, b\}, \{a\}\}$, hence the "EA representation"



does not make sense as one cannot use the same state differently.

The previous two examples raise the question of representability. In this paper we could only give a sufficient condition, the general case definitely would require non-trivial extra work.

Definition 2.5. The EA $\alpha : \Sigma \times Q \to Q$ is said to be *localizable* if for every state $q \in Q$ there is a parameter $v \in \Sigma$ such that $v \in q - \cup_{r \in Q, r \neq q} r$ (that is, v belongs only to the state q). Localizability means that every state has an eigenparameter, a parameter which is characteristic of the state. $\square$

In general, a state of a localizable EA can have many different eigenparameters, thus one rather speaks about the set of eigenparameters associated with a given state.

Proposition 2.6. Skeleta of localizable EA are isomorphic to EA.

Proof. The idea is that if α is localizable, then $\sim$ extends to a congruence of the state space Q. That is, if we let $Q/_\sim = \{q/_\sim : q \in Q\}$ where $q/_\sim = \{v/_\sim : v \in q\}$, then the automaton $\beta : \Sigma/_\sim \times Q/_\sim \to Q/_\sim$ defined by the equality

$$\beta(v/_\sim, q/_\sim) = \alpha(v, q)/_\sim$$

will be well-defined. As $Q/_\sim \subseteq 2^{\Sigma/\sim}$, β will be an EA.

By localizability for every state q there is a parameter v_q, an eigenparameter of q. For two distinct states q and r the corresponding eigenparameters cannot be $\sim$-congruent, because $\alpha(v_q, q) = q$ but q does not contain v_r, hence $\alpha(v_r, q) \neq q$. The same argument show that $\alpha(v_q, s) = q$ for every state $s \in Q$. Therefore all eigenparameters associated with a given state are equivalent, and each state can be identified with that equivalence class, that is, there is a bijection between Q and $Q/_\sim$. It follows that $\alpha(v/_\sim, q/_\sim) = \alpha(v/_\sim, q)$. Finally, $\sim$ is defined in such a manner that $\alpha(v, q) = \alpha(w, q)$ whenever $v \sim w$. Thus $\alpha(v/_\sim, q)$ is well-defined, hence β is well-defined, too. $\blacksquare$

Example 2.3 shows an EA which is not localizable (the state $[0, 1]$ does not have an eigenparameter), still its skeleton can be represented by an EA. This means that localizability is not necessary for being representable by an EA.

Representation of standard finite state automata can be understood (at least) in two different ways.

Definition 2.7. Let Ω be a finite alphabet and R a set of states. The FSA $\delta : \Omega \times R \to R$ is representable by an EA if there is EA α over Ω such that α and δ are isomorphic.

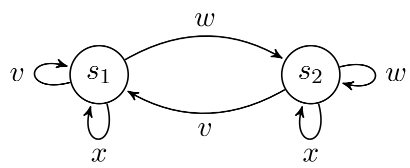
We say that α is representable in the general sense by an EA if there is a parameter space $\Sigma \supset \Omega$ and an EA α over Σ such that $\alpha \restriction \Omega$ is isomorphic to δ. $\square$

For an FSA $\delta : \Omega \times R \to R$ and a state $s \in R$ let us denote by $[s]_{\text{in}}$ the set $\{v \in \Omega : (\exists p \in R)\delta(v, p) = s\}$.

Proposition 2.8. Let $\delta : \Omega \times R \to R$ be an FSA with the property that there are no distinct states $s_1, s_2 \in R$ such that $[s_1]_{\text{in}} = [s_2]_{\text{in}}$. Then δ is representable by an EA.

Proof. The mapping $s \mapsto [s]_{\text{in}}$ is an isomorphism. ■

Example 2.9. Consider the following FSA over the alphabet $\{v, w, x\}$.



The sets of incoming edges $[s_1]_{\text{in}} = \{v, x\}$ and $[s_2]_{\text{in}} = \{w, x\}$ are different, thus after replacing s_i by $[s_i]_{\text{in}}$ we get an isomorphic Euclidean automaton.

Unfortunately, the condition in Proposition 2.8 is not necessary: one can construct an EA that does not satisfy that condition. Here is an easy example. Take a set S and a partition of S into non-empty sets S_1 and S_2. Take S to be the alphabet and put $Q = \{S, S_1, S_2\}$ as the set of states. The automaton is defined by $\alpha(v, q) = S$ for any $v \in S$ and $q \in Q$. Then $[S_1]_{\text{in}} = [S_2]_{\text{in}} = \emptyset$.

Connections with homomorphisms. In automata theory several different types of homomorphisms between automata are defined such as state-homomorphism, alphabet-homomorphisms, etc. Since states of Euclidean automata are subsets of the alphabet, there is a natural way to generalize these concepts: homomorphisms between Euclidean Automata was defined in Kornai (2014b) as follows.

Definition 2.10. A homomorphism from EA $\alpha : \Sigma \times Q \to Q$ to another EA $\beta : \Omega \times S \to S$ is a mapping $h : \Sigma \to \Omega$ such that the following stipulations hold.

- $h(\texttt{start}_\alpha) = \texttt{start}_\beta$;
- h extends to a mapping $h : Q \to S$ in the natural way;
- $h\big(\alpha(v_1, \ldots, v_n, \texttt{start})\big) = \beta(h(v_1), \ldots, h(v_n), \texttt{start})$.

□

By Proposition 2.6 skeleta of localizable EA remain Euclidean: For a localizable EA $\alpha : \Sigma \times Q \to Q$ the congruence $\sim$ defined by (1) extends to a congruence of the state space Q. That is, if we let $Q/_\sim = \{q/_\sim : q \in Q\}$ where $q/_\sim = \{v/_\sim : v \in q\}$, then the automaton $\beta : \Sigma/_\sim \times Q/_\sim \to Q/_\sim$ defined by the equality

$$\beta(v/_\sim, q/_\sim) = \alpha(v, q)/_\sim$$

is an EA. Let us denote this β by $\tilde{\alpha}$. It is very easy to check (cf. the proof of Proposition 2.6), that $\tilde{\alpha}$ and the skeleton $\bar{\alpha}$ defined in 2.2 are isomorphic. Therefore we will call $\tilde{\alpha}$ also a skeleton of α.

Now, if α is localizable, then $\tilde{\alpha}$ is a homomorphic image of α. For, write $h(v) = v/_\sim$, where $\sim$ is the congruence defined by (1). The first two items of Definition 2.10 follows from the proof of Proposition 2.6 and the third item is the very definition of $\tilde{\alpha}$ as $h\big(\alpha(v,q)\big) = \alpha(v,q)/_\sim = \tilde{\alpha}(v/_\sim, q/_\sim)$.

An important consequence is that localizable EA's are categorical objects in the sense that the class of all such automata is closed under the homomorphism introduced in Definition 2.10, and skeleta form a closed subcategory of the category of all localizable EA.

3. Languages accepted by EA

In this section we turn to a logical characterization of the languages that can be accepted by EA. Let $\delta : \Omega \times R \to R$ be a standard FSA. The language of α is the set $L_\alpha \subseteq \Omega^*$ defined as

$$L_\alpha = \{w \in \Omega^* : \ \delta(w, \texttt{start}) = \texttt{final}\}.$$

This definition clearly makes sense even if Ω is infinite. Therefore one can define without any difficulty when a Euclidean automata $\alpha : \Sigma \times Q \to Q$ accepts a language $L \subseteq \Sigma^*$: if and only if $L = L_\alpha$.

This definition, however, may not be satisfactory enough when Σ is infinite. The reason is that one might like to say that the skeleton of an EA accepts the same language as the original EA when restricted to the language of the skeleton. More precisely one can consider the skeleton acting on a subset of the original alphabet: pick a representative from each of the equivalence class of the alphabet of the skeleton. Then the skeleton and the original EA shows the same behavior on each input string. This motivates the next definition.

Definition 3.1. Suppose $\alpha : \Sigma \times Q \to Q$ is an EA, where Σ is allowed to be infinite. Let $\Omega \subseteq \Sigma$ be a finite subalphabet and $L \subseteq \Omega^*$ a language. Then α is said to accept L in the general sense if $L_\alpha \restriction \Omega^* = L$. □

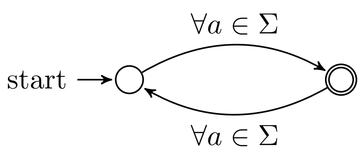
Proposition 3.2. Each regular language can be accepted in the general sense by an EA.

Proof. Let L be a regular language over Ω and $\delta : \Omega \times R \to R$ the FSA that accepts L. By adding new letters to Ω one can reach that in this expanded language (denote it by Σ) the FSA satisfies $[s]_{\text{in}} \neq [s']_{\text{in}}$ for all states $s, s' \in R$ by defining the action of the new letters carefully. The resulting FSA is representable by an EA α applying Proposition 2.8. It is clear that $L_\alpha \restriction \Omega^* = L$. The proof reveals that more is true: any language accepted by an FSA over a possibly infinite alphabet can also be accepted in the general sense by an EA. If Ω is finite then Σ can be chosen to be finite; otherwise they can have the same infinite cardinality. ■

It is obvious that every language accepted by an EA is regular (because EA are special FSA), thus the question of which languages are accepted in the general sense is settled. The cheat here is that we are allowed to enlarge the alphabet. Is it true that keeping the same alphabet, for every FSA $\delta : \Sigma \times R \to R$ there is a EA $\alpha : \Sigma \times Q \to Q$ such that $L_\delta = L_\alpha$? If the alphabet is finite, then the answer obviously is “no”. This is because over a finite alphabet Σ one can define at most finitely many EA as the set of states Q should be a subset of 2^Σ which is still finite. But what about the infinite case where one can have any finite number of states? The answer is still “no” but for different reasons: the language that contains words having odd length (over any alphabet Σ) can be accepted by an FSA but cannot be accepted by any EA.

Example 3.3. No EA can accept the language containing words having odd length.

Proof. The language $L = \{w \in \Sigma^* : |w| \text{ is odd } \}$ is accepted by the FSA figured below.



Suppose there is an Euclidean automaton (Q, I, F, α) that accepts L. The first input letter v_1 can be any letter from Σ, hence after the first transition $\alpha(v_1, \texttt{start}) = q_1$, the resulting state q_1 should contain v_1 by Euclideanity. This means $q_1 = \Sigma$. The second letter v_2 is also arbitrary, hence after the second transition $\alpha(v_2, q_1) = q_2$ we get similarly that $q_2 = \Sigma$. This means $q_1 = q_2$ and in fact continuing the argument one gets the conclusion that there can be only one state $Q = \{\Sigma\}$. But such an EA accepts all strings and not just the ones having odd length. ■

It is known that regular languages are exactly the languages that can be defined in monadic second order logic (Büchi 1960; Elgot 1961). Let us recall some of the basic definitions to make everything clear. Let Σ be an alphabet (possibly infinite) and let $w = \langle w_1, \ldots, w_n \rangle$ be a word in Σ^*. Such a word can be represented by the relational structure

$$\mathbf{w} = \big(\{1, \ldots, n\}, <^{\mathbf{w}}, (Q_v^{\mathbf{w}})_{v \in \Sigma}\big)$$

called the word model for w, where $<^{\mathbf{w}}$ is the usual ordering on the domain of w and $Q_v^{\mathbf{w}}$ are unary predicates collecting for each letter $v \in \Sigma$ those letter positions of w which carry v:

$$Q_v^{\mathbf{w}} = \{i : \; w_i = v\}$$

The corresponding first-order language $FO(\Sigma)$ has variables $x, y, \ldots$ and built up the grammar

$$\varphi ::= \; Q_v(x) \mid x < y \mid \neg\varphi \mid \varphi \vee \psi \mid \exists x \varphi$$

The language defined by a formula φ is $L_\varphi = \{w \in \Sigma^* : \; \mathbf{w} \models \varphi\}$, where the satisfaction relation $\models$ is defined in the usual way. For example the language where "every a is immediately followed by a b" can be defined by the formula

$$\forall x \big(Q_a(x) \; \to \; \exists y (y = x + 1 \; \wedge \; Q_b(y))\big)$$

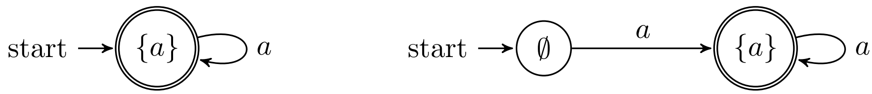
where $y = x + 1$ has the usual definition $x < y \; \wedge \; \neg\exists z (x < z \wedge z < y)$. A non-example could be $L = \{a^{2n} : \; n \in \mathbb{N}\}$ which is not expressible by a first order formula (Esparza 2012).

Monadic second order logic $MSO(\Sigma)$ is an extension of first order logic with variables $X, Y, \ldots$ ranging over sets of elements of models. The corresponding atomic formulas $X(x)$ are also introduced with the intended meaning "x belongs to X". Clearly $MSO(\Sigma)$ is more expressive than $FO(\Sigma)$ but not vice-versa as the next theorem shows:

Theorem 3.4 (Büchi 1960; Elgot 1961)**.** A language (over a finite alphabet) is recognizable by a finite state automaton if and only if it is $MSO(\Sigma)$-definable, and both conversions, from automata to formulas and vice versa, are effective.

Thus, regular languages are exactly the monadic second order definable languages. However, examples suggests that languages accepted by Euclidean automata are first order definable:

Example 3.5. For $\Sigma = \{a\}$ we must have $Q \subseteq \{\emptyset, \{a\}\}$ and thus there are exactly two non-isomorphic EA, figured below



The languages accepted by the automata are $L_1 = \{a^n : n \geq 0\}$ and $L_2 = \{a^n : n \geq 1\}$. Both languages are definable in the language $FO(\Sigma)$, respectively by the formulas $\forall x Q_a(x)$ and $\exists x Q_a(x) \wedge \forall x Q_a(x)$.

Example 3.6. For $\Sigma = \{a, b\}$ the number of variations is larger than before as $Q \subseteq \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$ gives more possibilities. We will not draw all the non-isomorphic EA's here, but one can check easily that the languages that can be accepted by EA over Σ are of the form "all sequences of a's and $b's$ + if we wish we can prescribe the first and last letter". For example such an L can contain all words starting with an a. In any case a $FO(\Sigma)$-characterization can be easily given. (As we prove next that languages accepted by EA are first order definable, we omit the details of the rather painstaking checking of this claim).

Indeed, we prove that languages accepted by Euclidean automata are FO-definable.

Proposition 3.7. For every finite alphabet Σ there is a corresponding first order language FO such that each language accepted by an Euclidean automaton can be defined by a FO-formula.

Proof. Assume (Q, I, F, α) is a Euclidean automaton over the finite parameter space Σ. Take the first order language $FO(\Sigma)$ described above and expand this language by new unary predicates R_i for $i < 2^{|\Sigma|}$. We have to find a first order formula that expresses in any given word model $\mathbf{w}$ that α accepts w. The formula in question will state the existence of a successful run of the EA. As α has finitely many state, we can enumerate them by $Q = \{q_i : i < 2^{|\Sigma|}\}$. Each state q_i of the EA will be encoded by the predicate R_i. We need to express the followings: (1) in each turn the machine can be only in one state; (2) the starting state is q_0; (3) if we are in position x and the next position is y, then we applied one of the letters, i.e. for one of the $v \in \Sigma$ we have $Q_v(x)$, and thus the next state should be the one prescribed by $\alpha(v, q_i) = q_j$; (4) the last position is a final state. Thus α accepts w if and only if

$$\begin{aligned} \mathbf{w} \models \quad & \Big(\bigwedge_{i \neq j} \forall x \neg (R_i(x) \wedge R_j(x)) \Big) & (6) \\ & \wedge \quad \forall x (\mathtt{first}(x) \to R_0(x)) & (7) \\ & \wedge \quad \forall x \forall y (x = y + 1 \ \to \bigvee_{\alpha(v,q_i)=q_j} (R_i(x) \wedge Q_v(x) \wedge R_j(y))) & (8) \\ & \wedge \quad \forall x (\mathtt{last}(x) \to \bigvee_{(\exists q \in F)\alpha(v,q_i)=q} (R_i(x) \wedge Q_v(x))) & (9) \end{aligned}$$

Since the empty word satisfies this sentence, if α does not accept the empty word, then a corresponding clause such as $\exists x(x = x)$ should be added. $\mathtt{first}(x)$ and $\mathtt{last}(x)$ are respectively the formulas $\neg\exists y(y < x)$ and $\neg\exists y(x < y)$. ■

If the alphabet Σ is not finite, then a similar argument shows that languages accepted by EA can be defined by first order formulas that are allowed to contain infinite disjunctions having an infinite vocabulary (i.e. we use the logic $FO_{\infty\omega}$).

Recall that *finiteness* is a property that cannot be expressed in first order logic. Indeed, by the compactness theorem if a formula holds in all finite models, then it should also hold in an infinite model. This can be seen as one of the main reasons why languages that can be accepted by finite state automata cannot be defined in first order logic (and one needs monadic second order logic). Even if the alphabet is fixed, an FSA can have an arbitrary finite number of states and we do not have any control, in terms of first order logic, over the number of states. As we have already seen, there are only finitely many EA over a finite alphabet. That is, if we fix the alphabet, then there is a fixed upper bound on the possible number of states, depending only on the size of the alphabet. This allows us to bypass the problem of non-definability of finiteness: using first order logic it is easy to define models having size at most n, for a fixed finite number n. This is the key for Proposition 3.7.

As we already mentioned, there are only finitely many EA over a finite alphabet. Therefore not every first order definable language can be accepted by an EA (there are infinitely many first order definable languages). Then what is the logic that is exactly as expressible as Euclidean automata? As the number of states is limited, the set of EA do not have any extensive closure property (such as closed under direct product, unions, etc). This suggests us the vague idea that EA are not logical in the sense of expressibility. Of course it is not clear how to define "logicality" in a precise manner.

Acknowledgements

I am grateful to the reviewer for his/her careful reading of the manuscript and the helpful suggestions. I wish to acknowledge the Premium Postdoctoral Grant of the Hungarian Academy of Sciences hosted by the Logic Department of Eötvös Loránd University.

References

Büchi, J. R. 1960. Weak second-order arithmetic and finite automata. Zeitschrift für mathematische Logik und Grundlagen der Mathematik 6. 66–92.

Elgot, C. C. 1961. Decision problems of finite automata design and related arithmetics. Transactions of the American Mathematical Society 98. 21–51.

Esparza, J. 2012. Automata theory, an algorithmic approach. Lecture notes.

Kornai, A. 2014a. Euclidean automata. In Implementing selves with safe motivational systems and self-improvement.

Kornai, A. 2014b. Finite automata with continuous input. In S. Bensch, R. Freund and F. Otto (eds.) Sixth Workshop on Non-Classical Models of Automata and Applications (NCMA 2014). Short papers. 19–24.

Sainsbury, R. M. and T. Williamson. 1997. Sorites. In B. Hale and C. Wright (eds.) A companion to the philosophy of language. Malden, MA & Oxford: Blackwell. 458–484.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 247–280

Thoughts on the semantics and pragmatics of rising declaratives in English and rise-fall declaratives in Hungarian

BEÁTA GYURIS
Research Institute for Linguistics,
Hungarian Academy of Sciences
gyuris.beata@nytud.mta.hu

KEYWORDS

rising declarative
rise-fall declarative
sentence type
felicity conditions
bias

ABSTRACT

The paper looks at the interpretation of a construction type in the Hungarian language referred to as *rise-fall declarative*, which is used to encode biased questions. Its felicity conditions are compared to those of rising declaratives in English, based on several recent accounts of the latter. It is argued that the theory proposed by Gunlogson (2003), complemented with two further conditions, can capture the licensing conditions of rise-fall declaratives in Hungarian correctly.

1. Introduction

Kálmán (2001, 101) discusses the interpretation of a type of “polar question” in Hungarian that bears a specific prosody, characterized by a rise-fall melody on all constituents in the *comment* part of the sentence (i.e., the part following the constituents in the *topic* field):[1]

[1] Cf. Kálmán (2001) and É. Kiss (2002) for a discussion of the syntactic structure of the Hungarian sentence. In the examples cited from Kálmán (2001) the original notation is retained. In the latter work, “^” marks a rise-fall contour, and “⊥” what the authors call a “pre-tone” on unstressed definite articles and relative pronouns, which do not constitute part of the preceding character-tone. The latter phenomenon will not be discussed in what follows. The last peak within the multiple rise-fall melody falls on the penultimate syllable of the sentence, just as it does in ordinary rise-fall interrogatives, illustrated below.

(1) ^Bekapcsolva hagytad a ^mobiltelefont a ^színházban?
switched.on left.2SG the mobile.ACC the theatre.in
'You left the mobile phone switched on in the theatre?' (Kálmán 2001, 101, (8))

Kálmán (2001, 101)[2] argues that as compared to ordinary "polar questions", this construction carries an interpretational surplus: "it suggests that the speaker, who presumably already knows the answer, only expects acknowledgement or explanation".[3] It is noted that this construction usually encodes echo questions uttered with a "disapproving attitude".[4]

It is claimed about a different example, shown in (2), that it illustrates a special case: the aim of the question is to help find out whether some state of affairs that the conversational participants can observe in the situation was brought about by the event described in the question.[5]

(2) ^Felhívtad ⊥a ^takarítónőt, ⊥akit ^ajánlottam? (Azért van ilyen rend?)
called.2SG the cleaner.ACC who.ACC recommended.1SG that.for be such order
'You called the cleaner I recommended? (Is that why it looks so orderly here?)'
(Kálmán 2001, 27, (8))

Often, the speaker asks these questions to express her surprise:

(3) A ^Lajos ^berakta ⊥a ^kefírgombát ⊥a ^mélyhűtőbe?
the Lajos VM.put.3SG the kefir.grains.ACC the freezer.into
'Lajos put the kefir grains into the freezer?' (Kálmán 2001, 27, (9))

A further illustration of the use of the same prosodic pattern is provided in (4), which appears to be an echo-question, repeating an utterance made before:[6]

[2] This publication was the output of a seminar taught by László Kálmán at the Theoretical Linguistics Program at ELTE Budapest. He contributed significantly to the individual chapters both in the course of the discussion and as editor of the volume. The examples cited here are from the chapters entitled *Kérdések* 'Questions' (pp. 98–135), written by Viktor Trón, and from the one entitled *A topik és a kontrasztív topik* 'The topic and the contrastive topic' (pp. 24–53), by Attila Novák and the present author.

[3] "...azt sugallja, hogy a beszélő, aki már feltételezhetően tudja a választ, csak megerősítésre, magyarázatra vár".

[4] "A konstrukciót általában rosszalló visszakérdezésként használjuk." (*ibid.*, 101)

[5] An analogous proposal about the use of the same prosody was previously made in Kálmán & Nádasdy (1994, 456).

[6] The conjecture about the context is mine, the authors supply no information about contexts in general.

(4) ^Gyanútlanul ^ment ^át ⊥az ^utcán, ⊥amikor ^egyszer csak
unsuspectingly went across the street.on when once only
⊥a ^fejére esett ⊥egy ^ablaktábla?
the head.his.onto fell.3SG an window.pane
'He was crossing the street unsuspectingly, when suddently a window pane fell onto his head?' (Kálmán 2001, 27, (10))

Varga (2010, 4) also discusses questions with a similar prosodic realisation, which he characterizes as follows: "each accent in the comment is retained, and the rise-fall can appear at every accented syllable, thus forming a sequence of repeated rise-falls". These forms encode, in his opinion, "a strongly incredulous, disbelieving yes-no question, which we ask in order to get some clarification of an unbelievable statement or experience". His example is shown in (5). (The prosodic notation follows that of Beckman & Pierrehumbert 1986.)

(5) L* H – L% L* H – L% L* H-L%
ˈMeghívták a ˈMelindát a ˈbulira?
VM.invited.3PL the Melinda.ACC the party.onto
'They have invited Melinda to the party? (How come?)' (Varga 2010, 4, (3b))

Both publications cited above are primarily concerned with the prosody and the felicity conditions of the construction under consideration, and do not go into issues of formal categorization. They both refer to the relevant construction as a type of "question".[7]

Our ultimate aim is to formally describe the felicity conditions of the construction type illustrated above. This paper wishes to contribute to

[7] Regarding the prosodic form of the construction, an anonymous reviewer of the current paper makes the observation that the pitch can either stay in the same range throughout the whole utterance or it can drift down with each phrase that is pronounced with a rise-fall. In the reviewer's opinion, the latter case seems to be characteristic of *incredulous questions* in general. My present suggestion is that it is not the downdrift that induces the appearance of incredulity. Instead, I assume that both phenomena are the result of information structural properties. Questions encoded by utterances containing multiple rise-falls tend to contain a lot of given material. (A question intended to ask for a specific piece of information in Hungarian can often take the form of a one-constituent utterance). One well-known reason for repeating given information is to call attention to some problem with it, e.g., the speaker's dissatisfaction. Given information tends to be pronounced with falling pitch cross-linguistically, which might explain the downdrift in this case. Although I consider these problems worthy of further attention, I will mostly disregard the issue of the downdrift in what follows.

this aim by looking at its formal properties, by comparing the contexts where it can appear to contexts where the construction referred to as *rising declarative* in Germanic and Romance languages is licensed, and by discussing to what extent the formal proposals that were put forward to capture the interpretation of the latter can be used to capture the felicity conditions of the Hungarian construction type under consideration. Section 2 first looks at the issue of how the sentence type of this prosodically marked construction can be determined.

2. On formal category membership

The construction type illustrated in (1)–(5) is used to encode question acts, which, as a default, is done by means of interrogatives in human languages. In Hungarian there are two ways of formally marking polar interrogatives: (i) by means of the *-e* particle, attached to the verb as a default ("-*e*-interrogatives"), illustrated below in (6a), and (ii) by means of a final rise-fall intonation ("/\-interrogatives"), whose peak falls on the penultimate syllable, illustrated in (6b). At first sight, it appears to be the case that the multiple rise-fall tones constitute an additional intonational "colouring" on ordinary /\-interrogatives and encode the emotional components of incredulity, disbelief, disapproval, or the fact that the questioner asks for acknowledgement, confirmation, or for an explanation for some state of affairs. In what follows, we will argue against such an approach, and for formally characterizing the sentence type under consideration as a *declarative*.

First, as noted in Gyuris (2017), whereas the polar interrogatives in (6a)–(6b) are perfectly grammatical if they contain negative polarity items (NPIs) like *valahol is* 'anywhere', the form with the multiple rise-fall tones is incompatible with NPIs, as shown in (6c):[8]

(6) a. Esik-e valahol is az eső?
falls-E anywhere too the rain
'Is it raining anywhere?' (Gyuris 2017, 5, (7b))

[8] An anonymous reviewer notes that (6c) is only ungrammatical on the non-downdrifted pronunciation of the utterance (cf. fn. 7). She/he considers all NPIs compatible with the downdrifted version of utterances bearing the multiple fall-rise prosody. Although a thorough discussion of the phenomenon has to wait for another occasion, it seems to me that the downdrifted version of (6c) should be analysed as an echoic /\-interrogative. Naturally, the validity of this suggestion can only be proven if the different pronunciations of (6c) are considered in the appropriate contexts.

b. Esik valahol is az eső/\?
'Is it raining anywhere?' (*ibid.*, 5, (7a))

c. *^Esik ^valahol is az ^eső? (*ibid.*, 6, (9))

Second, as also argued in Gyuris (2017), the Hungarian construction is incompatible with the pragmatic marker *vajon* 'I wonder', which both Kenesei (1992, 691) and Kálmán (2001, 99) claim to be restricted to the interrogative sentence type, and which therefore offers itself as a diagnostic property of interrogatives. The following examples illustrate the phenomenon:

(7) a. Have you been in touch with Mary lately?

b. Not at all.

a.′ Vajon talált-e már állást?
vajon found-E already job.ACC
'Has she already found a job, I wonder.'

a.″ Vajon talált már állást/\?
'Has she already found a job, I wonder.'

a.‴ *^Vajon ^talált már ^állást? (Gyuris 2017, 6, (10))

Third, as shown in Gyuris (2016; 2017), negative /\-interrogatives are ambiguous between so-called "inside" and "outside" negation readings (referred to as *IN* and *ON readings* in what follows), which were first discussed for English in Ladd (1981). (Cf. Büring & Gunlogson 2000, van Rooij & Šafářová 2013, Romero & Han 2004, Sudo 2013 for further analysis.) As also argued in Gyuris (2016; 2017), there are certain morphosyntactic features that are either only compatible with ON or only with IN readings of /\-interrogatives. If the construction type under consideration here were a /\-interrogative with an *additional* intonational marking, we would expect /\-interrogatives with an obligatory ON reading to have alternative pronunciations using the multiple /\-contour (abstracting away from contextual licensing conditions for a moment). This is not the case, however. (8)–(9) show that, as opposed to ordinary negative /\-interrogatives, questions encoded by the multiple rise-fall forms are not compatible with an *is* 'also' phrase, or with lack of inversion between prefix and verb following a negative particle *nem* 'not', which are both considered diagnostics of ON readings in Gyuris (2016; 2018):

(8) a. Nem ment el János is moziba/\?
not went VM János also movies.into
'Didn't John go to the movies too?'

b. *^Nem ment el ^János is ^moziba?

(9) a. Nem elment moziba/\?
not VM.went movies.into
'Isn't it the case that he went to the movies?'

b. *^Nem ^elment ^moziba?

Furthermore, (10a) and (10b) illustrate that whereas *vala*-indefinites can have both a specific and a non-specific reading in /\-interrogatives, in multiple rise-fall constructions they can only give rise to the former interpretation, just as they do in ordinary falling declaratives. (The relevant observation on the latter was made in Szabolcsi 2002, 220.)

(10) a. János nem hívott fel tegnap valakit/\?
John not called VM yesterday somebody.ACC
i. 'Didn't John call a particular person yesterday?
ii. 'Didn't John call some person yesterday?'
(Gärtner & Gyuris 2012, 401, (25), translations amended)

b. ^János ^nem hívott fel ^tegnap ^valakit?
i. 'John didn't call a particular person yesterday?'
ii. *'John didn't call some person yesterday?'

Fourth, we can see an interesting contrast between the compatibility of /\-interrogatives versus the multiple rise-fall constructions with certain speaker-oriented adverbs. (The observations were inspired by suggestions made by Abeillé et al. 2014 about adverbials in French rising declaratives.) (11a)–(11b) show that the adverb *esetleg* 'perhaps' is grammatical in the multiple rise-fall construction, but it is ungrammatical in an ordinary /\-interrogative:

(11) a. Esetleg ^bekapcsolva hagyta a ^mobiltelefont a ^színházban?
perhaps switched.on left.3SG the mobile.ACC the theatre.in
'He left perhaps the mobile phone switched on in the theatre.'

b. *Esetleg bekapcsolva hagyta a mobiltelefont a színházban/\?

c. Esetleg bekapcsolva hagyta a mobiltelefont a színházban.

The falling declarative in (11c) is also compatible with *esetleg*, which points to a similarity between multiple rise-fall constructions and declaratives.[9]

As far as the adverb *talán* 'perhaps' is concerned, the situation is even more interesting: it is compatible with both structures, but it leads to a rhetorical question interpretation in the case of ordinary /\-interrogatives (which the corresponding -*e*-interrogatives also share):

(12) a. Talán ^bekapcsolva hagyta a ^mobiltelefont a ^színházban?
perhaps switched.on left.3SG the mobile.ACC the theatre.in
'He left perhaps the mobile phone switched on in the theatre?'

b. Talán bekapcsolva hagyta a mobiltelefont a színházban/\?
'Did he perhaps leave his mobile phone switched on in the theatre?'
Intended meaning: 'He did not leave ...'

The formal and interpretational differences between /\-interrogatives and the multiple rise-fall-constructions encoding questions that we reviewed above indicate that the latter do not belong to the interrogative sentence type but to the declarative one. Therefore, in what follows, the form type under discussion here will be referred to as *rise-fall declarative*, abbreviated as /\-*declarative.*

In the next section we take a closer look at an intonationally marked declarative sentence type that is referred to in Germanic and Romance languages as *rising declarative.* We will contrast the felicity conditions of rising declaratives, discussed in the literature, to those of Hungarian /\-declaratives, make some observations on the validity of the theoretical proposals concerning the former, and put forward a modest proposal on how the licensing conditions of the latter can be modelled formally.

[9] An anonymous reviewer, while acknowledging the validity of the data in (11), notes that the negative counterpart of (11b), *Esetleg nem hagyta bekapcsolva a mobiltelefont a színházban/\?* is felicitous in Hungarian. Besides noting that the above structure can only give rise to an ON reading, I have no explanation for the phenomenon at the moment.

3. Formal analyses of rising declaratives and their possible applications to /\-declaratives

3.1. Gunlogson (2003) and a proposal for extending it

Gunlogson (2003) provides a range of new observations and a formal account in terms of context update semantics regarding the use of (rising and falling) declaratives and interrogatives to encode questions in English. For the sake of brevity, in what follows, we will focus on her claims concerning the contrasts between polar interrogatives (with inversion) and rising declaratives only.

The first among them is that whereas interrogatives are generally available to ask a question in an unbiased context, declaratives are not, as (13) illustrates:

(13) At a committee hearing:

a. Are you a member of the Communist party?

b. #You are a member of the Communist party? (Gunlogson 2003, 1–2, (5ab))

As (14) shows, a /\-declarative is unacceptable in Hungarian in the same context, too:

(14) At a committee hearing:

#Maga ^tagja volt a ^kommunista ^pártnak?
you member.its was the communist party.DAT
'You were a member of the communist party?'

Second, similarly to ordinary interrogatives, rising declaratives do not commit the speaker to the descriptive or propositional content of the declarative. They are felicitous even if the speaker is skeptical about the truth of the latter, as the following, *echoic* use illustrates:

(15) A and B are looking at a co-worker's much-dented car.
A: His driving has gotten a lot better.
B's response:

a. Has it? I don't see much evidence of that.

b. It has? I don't see much evidence of that. (Gunlogson 2003, 21, (44a–b))

The next example shows that a /\-declarative is also acceptable in the same context:

(16) A and B are looking at a co-worker's much-dented car.
A: His driving has gotten a lot better.
B's response:
B: Már ^sokkal ^jobban ^vezet? Nem sok jelét látom.
already much better drive.3SG not much sign.its.ACC see.1SG
'His driving has gotten a lot better? I don't see much evidence of that.'

The following example also illustrates lack of commitment by the speaker, but here it is not the propositional content, but the presuppositions of the interlocutor's utterance that are challenged:

(17) A: The king of France is bald.
B's response:

a. Is France a monarchy?
b. France is a monarchy? (Gunlogson 2003, 2, (7a–b))

Hungarian /\-declaratives are equally fine in the same context:

(18) A: The king of France is bald.
B's response:
^Franciaország ^királyság?
France monarchy
'France is a monarchy?'

To account for these and analogous data, Gunlogson derives the meaning and use of rising declaratives compositionally by proposing that the declarative/interrogative form and the rising/falling intonation introduce different types of context change potentials (CCP), which are then combined compositionally. According to this, whereas the declarative form marks the presence of commitment to the descriptive content of the sentence, the final rise signals that this commitment is attributed to the addressee and the fall signals that it is attributed to the speaker.

The formal model of the proposal uses, in addition to the concept of the Common Ground (Stalnaker 1978), the set of propositions representing the public beliefs or discourse commitments (DC) of the individual participants, referred to as DC_X for participant *X*, and the context set associated with each discourse commitment set, referred to as cs_X, which consists of the set of possible worlds compatible with the propositions in DC_X. Thus, the CCP of a declarative sentence is defined with respect to an individual cs_X, independently of the identity of *X*, as in (19), and the CCPs associated with rising and falling locutions as in (20)–(21), respectively, where C stands for the input and C′ for the output context:

(19) $cs_X + S_{decl} = \{w \in cs_X$: the descriptive content of S_{decl} is true of $w\}$
(Gunlogson 2003, 33, (74a–b))

(20) $C + \uparrow S = C'$ such that:

a. $cs_{Addr}(C') = cs_{Addr}(C) + S$
b. $cs_{Spkr}(C') = cs_{Spkr}(C)$ (*ibid.*, (75))

(21) $C + \downarrow S = C'$ such that:

a. $cs_{Spkr}(C') = cs_{Spkr}(C) + S$
b. $cs_{Addr}(C') = cs_{Addr}(C)$ (*ibid.*, (76))

Unifying the contributions of the declarative form and of the rise, the CCPs of rising declaratives look like as follows:

(22) $C + \uparrow S_{decl} = C'$ such that:

a. $cs_{Spkr}(C') = cs_{Spkr}(C)$
b. $cs_{Addr}(C') = cs_{Addr}(C) + S_{decl}$ (*ibid.*, (77))

(22) means that as a result of the utterance of a rising declarative, the context set of the speaker does not change, but that of the addressee does: only those possible worlds remain in it that are compatible with the descriptive content of S_{decl}.

Based on the above assumptions about the contribution of the declarative form and the rising tone, Gunlogson offers the following *Contextual Bias Condition* on declarative questions, which proposes that rising declaratives are only compatible with contexts where the addressee is publicly committed to the proposition expressed:

(23) Contextual Bias Condition
An utterance of S_{decl} with descriptive content p is interpretable as a polar question in C only if $cs_{Addr}(C) \subseteq p$. (Gunlogson 2003, 49, (105))

Let us now consider how the above proposal accounts for the examples illustrated above. First, the theory can easily predict why *echoic* (a.k.a. *reiterative*) uses of rising declaratives, as in (15), are felicitous: the interlocutor's previous utterance (echoed by the relevant S_{decl}) explicitly indicates commitment to the descriptive content of S_{decl}. Echoic uses of rising declaratives also subsume cases where the descriptive content of the S_{decl} corresponds to presuppositions of the addressee's previous utterance, as in (17), or to entailments of the proposition that the addressee has indicated commitment to. Both presuppositions and entailments are assumed to automatically enter the relevant interlocutor's cs. The following example appears problematic for this view since the proposition 'A talked to Helena' does not appear to be an entailment of the propositional content of A's utterance:

(24) A: Mark and Helena are leaving for Japan this week.
B: Oh ...

a. Did you talk to Helena?

b. You talked to Helena? (Gunlogson 2003, 56, (120))

Gunlogson argues, nevertheless, that (24b) can be accounted for along the same lines as the previous examples. If p is the descriptive content of the declarative question, and q is "a relevant public commitment of the Addressee's that serves as the basis for the inference" that p (*op.cit.*, 58), what is required additionally for this is that $q \rightarrow p$ be accommodated as a joint commitment of the participants.

Let us turn to rising declaratives that are used to encode *verification questions*. Whereas in a neutral context, illustrated in (25), only interrogatives but no rising or falling declaratives seem to be felicitous in English, in a context displaying evidence for the positive answer, as in (26), all these forms are acceptable.

(25) Robin is sitting in a windowless computer room with no information about current weather conditions when another person enters. Robin says to the newcomer:

a. Is it raining?

b. #It's raining?

c. #It's raining. (Gunlogson 2003, 60, (126))

(26) Robin is sitting, as before, in a windowless computer room when another person enters. The newcomer is wearing a wet raincoat and boots. Robin says:

a. Is it raining?

b. It's raining?

c. (I see that/So) It's raining. (Gunlogson 2003, 61, (128))

The Hungarian counterparts of (25)–(26), illustrated in (27)–(28), respectively, show that /\-declaratives have a distribution parallel to that of rising declaratives:

(27) Robin is sitting in a windowless computer room with no information about current weather conditions when another person enters. Robin says to the newcomer:

#^Esik az ^eső?
falls the rain
'It's raining?'

(28) Robin is sitting, as before, in a windowless computer room when another person enters. The newcomer is wearing a wet raincoat and boots. Robin says:

^Esik az ^eső?
falls the rain
'It's raining?'

Gunlogson accounts for the felicity of (26b) by saying that it satisfies the Contextual Bias Condition. There is public evidence that proposition *p*, denoted by the declarative, is true, which is thus accessible to the addressee. In her opinion, however, the addressee does not base his commitment to the truth of *p* on this evidence but on some other information he possesses due to his position, which he came by before the evidence became available to the speaker. The role of the public evidence for the addressee is that "it enables the Addressee to recognize that the Speaker is being intentionally uninformative", that is, that the speaker knows that the addressee knows that *p* is true (p. 62). The infelicity of (25b) is in turn attributed to the absence of publicly available evidence for the propositional content of the declarative. Gunlogson argues that even if Robin had access to information about the current weather conditions, unbeknownst to the newcomer, and "has good reason to be biased herself, together with the assumption that the Addressee is knowledgeable and may be presumed to have the same bias", this would not by itself improve (25b) (p. 82).

Interestingly, Gunlogson also adds the following remark to the discussion: "I want to deny that reiterative questions generally, and rising intonation specifically, are inherently associated with 'surprise' or 'incredulity', as is sometimes casually assumed." (p. 82) There is, however, one major problem, noted by Šafářová (2007, 305), which Gunlogson's account runs into. This concerns the apparent contradiction between the assumption that rising declaratives commit the addressee to the truth of the proposition in question and the fact that they normally still expect a response from the addressee. This problem will be addressed by the theories discussed below.

Let us now consider the possibility of adopting Gunlogson's theory for /\-declaratives in Hungarian. Echoic uses, like the one in (16), satisfy the Contextual Bias Condition (referred to as CBC in what follows). The addressee is committed to the propositional content of his utterance, as well as to the latter's presuppositions and entailments, thus, if the descriptive content of the echoic /\-declarative is identical to one of these, the utterance of the declarative is licensed.

I want to propose, however, that in addition to the satisfaction of the CBC, there is a further necessary condition on the use of /\-declaratives, which is based on the following intuition: although the speaker must (similarly to the addressee) be committed to the publicly available evidence, she cannot have a commitment to p before the evidence became available in the context. This is based on a general requirement on question acts, captured in Searle's Preparatory Condition 1 (Searle 1969, 66), which also plays a role in later studies of rising declaratives, discussed below. The new set of felicity conditions of /\-declaratives are shown below in (31).[10] The formula uses the abbreviation $\mathrm{cs}_{\mathrm{Spkr}}(\mathrm{C}^{-1})$, which refers to the context set of the speaker in the stage of the context that preceded the one in which the /\-declarative was uttered.

(29) Felicity conditions of /\-declaratives in Hungarian (to be revised)
A /\-declarative $\mathrm{S}_{\mathrm{decl}}$ with propositional content p is felicitous in a context C only if
a. $\mathrm{cs}_{\mathrm{Addr}}(\mathrm{C}) \subseteq p$. (Contextual Bias Condition, Gunlogson 2003, 49, (105))
b. $\mathrm{cs}_{\mathrm{Spkr}}(\mathrm{C}^{-1}) \nsubseteq p$.

(29) thus proposes that the utterance of /\-declaratives is only felicitous if they satisfy the CBC, and if the speaker was not committed to the propositional content of the /\-declarative before.

As far as /\-declaratives used as verification questions, as in (27)–(28), are concerned, felicitous occurrences also obey the conditions in (29). In the case of (28), there is publicly available evidence (which the addressee is thus supposed to be committed to) that supports the truth of the propositional content p of the declarative, the speaker appears to utter the question as a reaction to this evidence, and she is not assumed to be committed to the proposition 'It is raining' before. In the context of (27), the first condition is not satisfied, which explains it infelicity.

Let us consider now the Hungarian counterpart of (24), encoded by a /\-declarative:

[10] The assumption that /\-declaratives are only licensed if the speaker is not committed independently of the evidence to the propositional content is also inspired by the definition of *compelling contextual evidence* in Büring & Gunlogson (2000). Cf. Gyuris (2017) for further discussion of the role of compelling contextual evidence in the licensing of polar interrogatives available for encoding question acts in Hungarian.

(30) A: Mark and Helena are leaving for Japan this week.
B: Ó, ...
$^\wedge$Beszéltél $^\wedge$Helénával?
talked.2SG Helena.with
'You talked to Helena?'

I believe that the acceptability of the /\-declarative in (30) can be explained if we assume that, other things being equal, whenever there is public commitment to a proposition q that entails, on the speaker's judgment, proposition p, a /\-declarative with propositional content p is licensed. This condition can be integrated into (29) as illustrated in part (a-ii) of (31a), where $B_{\text{Spkr}}(\varphi)$ is a shorthand for 'Speaker believes that φ'. The resulting criterion for the felicitous occurrence of /\-declaratives will be referred to as the Interlocutor Bias Condition (IBC) in what follows. This replaces the condition proposed in (29):

(31) Interlocutor Bias Condition (IBC)
A /\-declarative S_{decl} with propositional content p is felicitous in a context C only if
a. there is a proposition q such that
(i) $\text{cs}_{\text{Addr}}(\text{C}) \subseteq q$, and
(ii) $B_{\text{Spkr}}\ (q \rightarrow p)$
b. $\text{cs}_{\text{Spkr}}(\text{C}^{-1}) \nsubseteq p$.

(31) states that the necessary conditions for the licensing of a /\-declarative in a context C include the following: there should be a proposition q that the addressee is committed to (clause a-i), which, according to the speaker's beliefs, entails p (clause a-ii), and the speaker was not previously committed to p (clause b). Let us consider how (31) accounts for the felicity of (30). At first sight, it seems reasonable to assume that q equals the propositional content of A's utterance, 'Mark and Helena are leaving for Japan this week'. However, I do not think that B could be attributed a belief according to which the latter entails the proposition 'A talked to Helena'. What the proposition 'A talked to Helena' can be assumed to follow from is the proposition q = 'A made an assertion whose propositional content equals 'Mark and Helena are leaving for Japan this week' '. Since this q describes a publicly available evidence, B has to be committed to it, and she can reasonably attributed the belief that $q \rightarrow p$, on the basis of Searle's Preparatory Condition 1 of Assertions (Searle 1969, 66), and on the basis of the fact that A observes Grice's Communicative Principle. Note that as opposed to Gunlogson's comments on (24), $q \rightarrow p$ does not have to be a joint belief of speaker and addressee (since the addressee does not have to have a belief that $q \rightarrow p$), which is reflected in (31).

I want to turn back to echoic uses of /\-declaratives and show that they can be assimilated to verification questions. This is based on the strategy we applied in the case of (30) above. Instead of taking the q in (31) to be the propositional content of the utterance made by the addressee, we take it to be identical to the following proposition: 'Hearer made an utterance of an assertion/question/etc. using sentence S with descriptive content φ'. Then, (31) amounts to saying that there is a belief by the speaker that the previous utterance, based on the felicity conditions of the respective speech acts entail the propositional content of the /\-declarative.[11]

The last example by Gunlogson discussed here (translation of the corpus example of Beun 2000 from Dutch) is shown below in (32). Here the last question by B is considered felicitous by the author:

(32) A: Schiphol Information
B: Hello, this is G.M. I have to go to Helsinki, from Amsterdam. Can you tell me which flights leave next Sunday?
A: Just a moment.
A: Yes, there are several flights. One leaves at 9.10, one at 11.10, and one at 17.30.
B: The flight takes about three hours? (Gunlogson 2003, 58, (123))

Gunlogson claims that in spite of the fact that the CBC does not seem to be fulfilled in the context of (32) (since the propositional content of the rising declarative is not logically entailed by the public commitments of the addressee), the felicity of the last utterance of B can be accounted for by assuming that the rising declarative is *accommodated* as a question, "by making the necessary contextual adjustment to meet the Contextual Bias Condition". Since "[t]here is no particular commitment of the Addressee's from which the content of the declarative is taken to follow", the author suggests that there is "a kind of blanket accommodation available for any declarative content presented by B that pertains to A's acknowledged area of expertise, i.e., airport information".[12] As a result, "when B presents p to A declaratively, it can be taken to follow from mutual assumptions that A already knows, or is in a position to confirm, p" (p. 59).

[11] Note that taking q to be identical to the proposition 'Hearer made an utterance of an assertion/question/...using sentence S with descriptive content φ' does not influence clause (a-i) of (31), since the former describes contextual evidence, which both interlocutors, including the addressee, must be committed to.

[12] According to her, the accommodation relies (i) on A being "mutually understood to be possessed of facts about some particular domain" (i.e., airport operations), and (ii) on the fact that "B has reason to believe that some proposition p is a fact and that A knows it by virtue of" (i) (i.e., "B believes p to be a fact about airport operations").

Independently of whether this account correctly captures the facts pertaining to English, the Hungarian counterpart of (32) containing a /\-declarative is not licensed in the relevant context:

(33) Same context as in (32)
B: #^Három ^órás az ^út?
three hourly the way
'The flight takes three hours?'

The infelicity of (33) suggests to me that what Gunlogson refers to as the "blanket accommodation" does not apply in the case of the Hungarian construction. Clause (a) of the IBC, however, does predict that (33) is not felicitous in the context: the hearer is not publicly committed to the truth of any proposition q about which the speaker could believe that it entails the proposition 'The flight takes three hours.'

In the next section we look at Poschmann (2008), which raises several points of criticism against Gunlogson's proposal.

3.2. Poschmann (2008)

Poschmann (2008) disagrees with the approach proposed in Gunlogson (2003), according to which declaratives used as questions uniformly involve a commitment shift from speaker to addressee. She proposes instead that the two classes of declarative questions that Gunlogson accounts for in a uniform fashion, namely, echo questions, in (15) and (17) above, and verification questions, referred to by Poschmann as *confirmative questions,* in (26b), should be given different treatments.

Poschmann (2008, 252) argues that whereas "utterers of echo questions can easily dissociate themselves from the content of their utterance, utterers of confirmative questions obviously cannot" . The contrast can be illustrated with the help of (34) vs. (35):

(34) Echo question:
A: Don't worry. The manager has of course been informed.
B: The manager has of course been informed? I wouldn't expect that.
(Poschmann 2008, 252, (9))

(35) Confirmative question:
At Tim's graduation. Tim's standing next to Sophie, a woman in her sixties.
Jack to Sophie: You're Tim's mother? (*I don't believe so.)
(Poschmann 2008, 257, (19), context description slightly amended)

As the following examples show, judgments are similar concerning the Hungarian counterparts of the examples above:

(36) A: Ne aggódj. Az igazgatót már természetesen tájékoztatták.
not worry.SUBJ.2SG the director.ACC already naturally informed.3PL
'Don't worry. The manager has of course been informed.'

B: Az ^igazgatót ^már ^természetesen ^tájékoztatták? Nem hiszem.
the director.ACC already naturally informed.3PL not believe.1SG
'Themanager has of course beeninformed? I don't believe it.'

(37) At Timi's graduation. Timi's standing next to Sophie, a woman in her sixties. Jack to Sophie:
^Ön a ^Timi ^édesanyja? (*Nem hiszem.)
you the Timi mother.her not believe.1SG
'You're Timi's mother? (*I don't believe so.)'

Poschmann proposes that "[i]n contrast to echo questions, confirmative questions seem to convey speaker commitment even though they certainly do not express the speaker's full beliefs" (*op.cit.*, 252). This observation is supported by the fact that confirmative questions seem to be possible both with rising and falling intonation cross-linguistically (Dutch, English), whereas the intonation of echo questions is obligatorily rising. She notes, in addition, that echo-questions and echo-assertions do not necessarily attribute commitment to the addressee, they can attribute it to a third person as well, and thus suggests that the commitment shift they involve is not connected to questionhood but to their being meta-representations.

Given the contrasts listed above, Poschmann (2008) argues that echo questions and confirmative questions constitute two different types of speech acts: the former involve commitment shift, and the latter speaker commitment. The speaker's commitment "depends on the addressee's acknowledgement: the speaker's commitment is suspended as soon as the addressee denies the content" (*op.cit.*, 257).

After showing that a classical implicature-based theory cannot handle confirmative questions properly,[13] Poschmann proposes an account of them that follows Zeevat (1996) and Nilsenová (2001) in assuming that "the

[13] Such a theory would attribute the fact that Jack's utterance in (35) should be interpreted as a question to an implicature: Jack's assertion stating his assumption about Sophie being Tim's mother would be completely uninformative for Sophie, thus it would violate the Maxim of Quantity. Thus, instead of being interpreted as an assertion, the utterance is interpreted as a question. The reason why Poschmann rejects this approach is that although the proposition that Sophie is Tim's mother is

speaker's utterance does not bring an update (of the Common Ground) with the proposition p expressed by the utterance but rather with the proposition $B_{Spkr}(p)$ – the speaker believes that p. In order for the proposition to become part of the Common Ground (that means a commitment of both speaker and hearer), the hearer has to acknowledge it, with the update $B_H(p)$" (Nilsenová 2001, 34).

Poschmann (2008, 258) argues that whereas the context conditions are usually sufficient to trigger this acknowledgement, it is the rising intonation that explicitly realizes the call on the addressee. She proposes that "rising intonation in speech acts involving speaker-commitment can be used to ask explicitly whether the addressee acknowledges the speech act performed by the speaker". Thus, rising intonation lends speech acts with speaker commitment a specific tentative reading, since it triggers a supplementary question about the acknowledgement (Ack) of the speech act performed (F(p)). This strategy boils down to treating confirmative questions not as questions but as complex speech acts consisting of an assertion and a question, represented formally as follows:[14]

(38) ASSERT (speaker, p) + QUEST (speaker, addressee, (Ack (F (p))))
(Poschmann 2008, 258, (21))

In Poschmann's opinion, the acknowledgement of an illocutionary act involves two steps: (i) the addressee acknowledges his understanding of the speech act, and (ii) the addressee accepts or refuses the content of the speech act. Rising intonation can be used to ask for both kinds of acknowledgement, the content disambiguates as to which of them is relevant in the situation. In the case of confirmative questions like (35) above, where the addressee, not the speaker, is the source of information concerning the truth of the propositional content of the utterance, acknowledgement of type (ii) plays a role. As far as "informative rising declaratives", such as the one illustrated in (39), are concerned, acknowledgement of type (i) plays a role (given that the speaker is an authority as far as her place of origin is concerned).[15]

uninformative for Sophie, it is not uninformative for the Common Ground, thus the implicature does not go through after all.

[14] (38) looks very similar to the interpretation Reese and Asher (2006) attribute to nuclear tag questions. The question therefore arises what predictions the present approach makes about the interpretational difference between rising declaratives and tag questions, which does exist, since the two forms cannot be replaced by each other freely in any context.

[15] The example originates from Hirschberg & Ward (1995), and is also mentioned in Gunlogson (2003), but is left without specific consideration there.

(39) Informative rising declarative:
Radio station DJ: Good morning, Susan. Where are you calling from?
Caller: I'm from Skokie? (Poschmann 2008, 259, (23))

As far as the assertion part of the complex speech act encoded by confirmative questions according to (38) is concerned, Poschmann does not seem to assume any specific licensing conditions: the speaker commitment required for an assertion to be felicitous can either be based on contextual evidence or on the private assumptions of the speaker. The fact that the latter suffices in her opinion is demonstrated by her comments on (25), repeated below in (40). She claims that if there is a hint in the context that there was a source for a private assumption, e. g. , the internet, the rising declarative in (40b) sounds much less problematic than without it.

(40) Robin is sitting in a windowless computer room with no information about current weather conditions when another person enters. Robin says to the newcomer:

a. Is it raining?

b. #It's raining?

c. #It's raining.

Although I do not want to deny that in a context modified the way suggested by Poschmann (40b) would be acceptable, it must be noted that this context would not meet the description given in (40) itself: Robin would *not* be without any information about current weather conditions. In any case, the previous discussion shows that Poschmann's views on the acceptability of rising declaratives in a situation where only the speaker's private information supports the truth of the propositional content are in opposition to those put forth by Gunlogson (2003).

The felicity of the last utterance in (32), repeated in (41) below, accords with Poschmann's general assumptions, and is predicted in terms of (38), without having to invoke an extra mechanism as Gunlogson does: speaker B makes an assertion that is based on her private assumptions, and at the same time asks for acknowledgement about the latter's content.

(41) A: Schiphol Information
B: Hello, this is G.M. I have to go to Helsinki, from Amsterdam. Can you tell me which flights leave next Sunday?
A: Just a moment.
A: Yes, there are several flights. One leaves at 9.10, one at 11.10, and one at 17.30.
B: The flight takes about three hours?

Although (38) seems to account well for the examples discussed here, I am not convinced that taking the utterance of rising declaratives to amount to the assertion of their propositional content (in addition to asking for the addressee's opinion about it), as the formula suggests, is what the account by Nilsenová (2001), which Poschmann takes as inspiration, suggests. In particular, (13) or (41) illustrate cases where the speaker's commitment seems to be much weaker than what is required for an appropriate assertion.

Importantly, it should be noted that the Hungarian version of (39), containing a /\-declarative, is infelicitous:

(42) Radio station DJ: Good morning, Susan. Where are you calling from?

Caller: #^Karcag ^mellől ^telefonálok?
Karcag beside.from call.1SG
'I'm calling from the area of Karcag?'

The fact that /\-declaratives are not licensed to appear in the context of (42) suggests that acknowledgement type (i) does not play a role in the interpretation of /\-declaratives in Hungarian. The infelicity of the /\-declarative is predicted, however, on the basis of clause (a) of the IBS: in the input context the addressee is not committed to any proposition q such that the speaker could reasonably be assumed to believe $q \rightarrow$ 'Speaker is calling from the area of Karcag'.

Interestingly, there is a form in the Hungarian language that can be used to make a speech act analogous to that in (39): this is a declarative pronounced with a final rising tone, illustrated in (43):

(43) Caller: Karcag mellől telefonálok/.
'I'm calling from the area of Karcag?'

The discussion of the felicity conditions of these "Hungarian rising declaratives" will, however, be left for a further occasion.

Rising declaratives that are said to involve an acknowledgement of the content of the speech act in Poschmann's framework, exemplified by (13), (15), (17), and (26b) above, and those involving an acknowledgement of the form, as in (39), are also distinguished in Jeong (2017)'s framework.[16] The author refers to the former type as *inquisitive* rising declaratives and to the latter as *assertive* (or informative) rising declaratives. Jeong argues that the two classes should be associated with different felicity conditions in English. As discussed above, assertive rising declaratives do not have /\-declarative counterparts in Hungarian. There are, however, inquisitive

[16] The journal article Jeong (2018) that developed from Jeong (2017) appeared too late to be given a proper discussion here. This will have to wait till another occasion.

rising declaratives, such as (41), as well, that cannot be translated into Hungarian with the help of /\-declaratives, as shown in (33). These contrasts also suggest that the felicity conditions proposed to account for the former cannot be appropriate for the analysis of the latter.[17]

In the next section we turn to the proposal by Gunlogson (2008), which was put forward partly as a reaction to Poschmann's criticism, and incorporates some new insights.

3.3. Gunlogson (2008)

Gunlogson (2008) revises the author's previous proposal, partly in order to be able to account for the data that Poschmann (2008) found incompatible with it. In order to explain the felicity of the relevant examples, Gunlogson proposes in this more recent paper that declaratives used as questions are acceptable in a context where they independently satisfy (i) felicity conditions that are associated with the use of declaratives, and (ii) conditions on the context that make the questioning interpretation possible, which is facilitated by the rising tone.

In the new framework, both rising and falling declaratives are claimed to encode the speaker's commitment, which, however, does not have to rely on contextual evidence, but can also be based on private information. The felicity conditions of initiating (that is, non-echo) uses of declarative questions rely on the concept of *sourcehood:*

[17] (i) illustrates an inquisitive rising declarative that cannot be encoded by a /\-declarative in the same context, as (ii) shows.

(i) A: The queen will arrive in five minutes.
B: O.K. The manager has of course been informed? *I wouldn't expect that.
(Poschmann 2008, 252, (10))

(ii) A: The queen will arrive in five minutes.
B: O.K. #Az ^igazgatót ^már ^természetesen ^tájékoztatták?
'O.K. The manager has of course been informed?'

The infelicity of (ii) comes as a surprise, since the example appears analogous to (30): if the speaker can be attributed a belief that connects the evidence and the propositional content of the /\-declarative, the utterance of the latter should be licensed in the context. Without looking deeper into possible ways of differentiating between the two contexts, I want to argue that the infelicity of (ii) can be attributed to the form of the declarative, particularly, the presence of *természetesen* 'of course', which cannot appear in utterances used to make question acts, other than those encoding echo questions.

(44) An agent α is a source for a proposition φ in a discourse d iff:

a. α is committed to φ; and

b. according to the discourse context, α's commitment to φ in d does not depend on another agent's testimony that φ in d. (Gunlogson 2008, 113, (27))

Assuming that all commitments have sources (referred to as the *Source Principle* in Gunlogson 2008, 117), the initiating uses of declarative questions satisfy the *Rule of Initial Commitment,* defined as follows:

(45) Rule of Initial Commitment

A speaker making a discourse commitment to φ in a context neutral with respect to φ is expected to be a source for φ. (Gunlogson 2008, 118, (39))

Gunlogson accounts for the infelicity of the declarative questions in (25), repeated again in (46), in the new framework as follows:

(46) Robin is sitting in a windowless computer room with no information about current weather conditions when another person enters. Robin says to the newcomer:

a. Is it raining?

b. #It's raining?

c. #It's raining.

Using a declarative involves commitment by the speaker, which makes the speaker to be the expected source of the commitment. However, "according to what is known about Robin's resources in the discourse situation, she is not a plausible source." As a result, "Robin's intention in uttering the declarative is unrecognizable, resulting in infelicity" (Gunlogson 2008, 119).

The situation in (26), repeated in (47), differs from the latter in that "it gives Robin a visible basis for her commitment", and it makes it "conceivable in the context that Robin could reach the conclusion that it's raining without the newcomer telling her" (*idem.*):

(47) Robin is sitting, as before, in a windowless computer room when another person enters. The newcomer is wearing a wet raincoat and boots. Robin says:

a. Is it raining?

b. It's raining?

c. (I see that/So) It's raining.

Although the evidence that the speaker bases her commitment on is present in the dicourse context of (47), Gunlogson argues that it is not necessarily the case, as (32) or (35), repeated in (48), illustrates. "[W]hat is generally required for felicity of a declarative is just that the discourse context allow the inference that the speaker has some basis for her choice" (*ibid.*, 120).

(48) Confirmative question:
At Tim's graduation. Tim's standing next to Sophie, a woman in her sixties.
Jack to Sophie: You're Tim's mother? (*I don't believe so.)

She argues with respect to (48) that "there is no particular evidence that the woman standing next to Tim is his mother", and "though the basis for Jack's conjecture might be partly or entirely contextual (Sophie's proximity to Tim, say, together with the favorable odds of encountering a parent at graduation), the declaratives seem to work without requiring us to make that assumption" (Gunlogson 2008, 105). I believe, however, that without these contextual bases no speaker would be considered justified to utter the rising declarative in (48) or its Hungarian counterpart in (37), which means that their felicitous uses are covered by the CBC and the IBC, after all.

The proposal according to which the speaker, who commits as source, must have adequate evidence (otherwise infelicity arises) explains only why the declarative form is felicitous in English. To account for why the declarative form can give rise to the questioning interpretation the author puts forth the following condition:

(49) Contingent Commitment Criterion
An utterance of a declarative with content φ is questioning to the extent that the speaker's commitment is understood as contingent on the addressee's ratification of φ. (Gunlogson 2008, 129, (48))

The role of the rising intonation is then seen by the author as marking the utterance as contingent "on some discourse condition whose identity is determined in context" (*ibid.*, 29).

Gunlogson looks at the infelicitous example (50b):

(50) (To coworker eating a piece of fruit.)
a. Is that a persimmon?
b. #That's a persimmon?
c. #That's a persimmon. (Gunlogson 2008, 102, (3))

She claims that it appears reasonable to assume that "the speaker has some private basis for thinking the fruit might be a persimmon" (*ibid.*, 131), thus the condition according to which commitments made with the help of declaratives must have sources (i.e., the Source Principle) is not violated. However, the Contingent Commitment Condition does seem to be violated, since there is no indication in the context that the addressee is acquainted with the name of the fruit he consumes. People are generally aware of the name of the food they eat but this follows from a generalization about people and not from properties of the context.

Note that besides echoic uses of rising declaratives, the account does not apply to assertive rising declaratives, as in (39), either, since they violate the Contingent Commitment Criterion. This means that Gunlogson's new theory is restricted to a smaller set of data than any of the previous approaches.

As (51) illustrates, the Hungarian /\-declarative counterpart of (50b) is equally infelicitous in the same situation:

(51) (To coworker eating a piece of fruit.)
#^Ezt hívják ^datolyaszilvának?
this.ACC called persimmon.DAT
#'This is called persimmon?'

Although it might follow from the common ground or from the properties of the situation that people in general or the addressee in particular knows the name of the food he is eating, it does not follow from either that the name of the fruit is a persimmon. Thus, there is no contextual evidence in the context that the addressee could be said to be committed to, and which is such that a reasonable speaker could believe it to entail the proposition 'This is called persimmon'. As a result, the infelicity of (51) is easily accounted for in terms of clause (a) of the IBC.

Consider next (52), where the appearance of a rising declarative is licensed, and its Hungarian counterpart containing a /\-declarative, in (53):

(52) (Laura has just entered the room, where Max sees her for the first time that day.) Max:
a. Did you get a haircut?
b. You got a haircut? (Gunlogson 2008, 104, (8))

(53) (Laura has just entered the room, where Max sees her for the first time that day.) Max:
^Levágattad a ^hajad?
VM.have.cut.1SG the hair.your
'You got a haircut?'

Gunlogson claims that the felicity of (52) is accounted for successfully on the basis of the Rule of Initial Commitment and the Contingent Commitment Criterion, since the "contingency of [Max's] commitment upon Laura's authority is inferable in the discourse context" (*op.cit.*, 129). I believe that the IBC also explains (53). It is without doubt that clause (b) of the IBC holds for (53): Max was not committed before seeing Laura to the truth of her having had a haircut. Let us assume that, according to clause (a) of the IBC, there is a proposition q in the context that describes the way Laura looks, which Laura herself is committed to (given that she is committed to public evidence). If the speaker can be attributed the belief that q entails the proposition 'Laura got a haircut', the felicity of (53) follows . The next section turns to the proposal by Malamud and Stephenson (2015).

3.4. Malamud & Stephenson (2015)

The proposal by Malamud & Stephenson (2015) regarding the interpretation of rising declaratives in English is based on two assumptions, which seem to be inspired by the theories of Gunlogson (2008) and Poschmann (2008), and thus the account seems to be the unification of the latter two. They argue, first, that rising declaratives introduce *projected* (rather than present) commitments by the speaker, which remind one of Gunlogson's contingent commitments, and second, that these constructions add a context-dependent *metalinguistic issue*, which needs to be resolved, and which reminds one of Poschmann's (2008) suggestion. Malamud and Stephenson represent the interpretation of rising declaratives in the framework proposed by Farkas and Bruce (2010), whose main features are summarized below.

Farkas and Bruce (2010, 85) make use of a (possibly empty) set DC_X for each participant X, consisting of the propositions that "X has publicly committed to during the conversation up to the relevant time, and which are not shared by all the other participants", a set CG of propositions shared as joint discourse commitments by all participants, a stack of sentential form/meaning pairs called *Table,* and a set PS ("projected set") of "projected" or "privileged" future common grounds. "The *Table* records what is 'at issue' in the conversation. When the Table is not empty, the immediate goal of the conversation is to empty it, that is, to settle the issue at hand. [...] A conversation is in a stable state when its Table is empty" (*ibid.*, 87).

The system above is enriched by Malamud and Stephenson in two respects. First, they add *projected commitments,* "things that interlocutors are expected to become committed to in the normal course of conversation" (*ibid.*, 299), which thus "represent the expected next stage of the conversation" (*ibid.*, 288). A projected commitment of the speaker or hearer will turn into an actual commitment after the hearer has confirmed it. The authors emphasize the specific nature of projected speaker commitments, "given that the speaker is always in full control of her own commitment set" (*ibid.*, 288). Thus, "if the speaker chooses to make a projected commitment, rather than a present one, the hearer(s) can infer that the speaker has some reason to delay making a commitment that she would otherwise be willing to make" (*idem.*).[18]

Second, they add the option of introducing a *metalinguistic issue,* and propose that the rising tone signals the existence of such an issue. When a rising declarative is uttered, both its propositional content p and then a (possibly singleton) set of propositions MLI_p is added to the *Table.* MLI_p is a "contextually determined set of propositions, any of which would resolve the contextually determined metalinguistic issue concerning p" (Malamud & Stephenson 2015, 296). A move that simultaneously involves a commitment and a metalinguistic issue indicates to the hearer that the commitment is a projected one, pending the resolution of the metalinguistic issue. "Any aspect of the utterance's content and form can be the subject of an *MLI,* as long as the speaker can give the hearer enough clues about its nature" (*idem.*).[19] Rises, therefore, are "predicted to be possible whenever the speaker isn't sure if a plain assertion is appropriate" (*idem.*).

In what follows, we illustrate the procedure with some examples given by the authors, also showing the corresponding Hungarian examples encoded by /\-declaratives. In (54)–(57) the metalinguistic issue concerns the correctness of an inference by the speaker based on the interlocutor's utterance:

(54) A and B are gossiping. A doesn't know anything about B's neighbor. B says, blushing, "You've GOT to see this picture of my new neighbor!" Without looking, A replies:
A: He's attractive? (Malamud & Stephenson 2015, 279, (2c))

(55) A: ^Jól néz^ki?
well look VM
'He looks good?'

[18] Importantly, the system including projected commitments for each participant differs from Gunlogson's contingent commitments in that Malamud and Stephenson also include projected hearer commitment, which lacks a counterpart in Gunlogson (2008).

[19] This feature of the proposal was inspired by Ginzburg (1996; 2012).

(56) A and B are gossiping. A doesn't know anything about B's neighbor. B says, blushing, "You've GOT to meet my new neighbor!" A replies:
A: He's single? (Malamud & Stephenson 2015, 280, (5c))

(57) A: ^Nőtlen a ^szomszédod?
single the neighbour.your
'Your neighbour is single?'

In (54) and (56), A infers that the neighbor is attractive or single, respectively, only indirectly; the metalinguistic issue concerns the question of whether the speaker's inference regarding the hearer's blushing is correct. (55) and (57) show that the corresponding Hungarian examples are felicitous in the same situations. These data would be explained by the IBC as well, if the speaker can be taken to believe that it follows from the fact that the interlocutor made the preceding utterance that the neighbour is good-looking or single, respectively. Note the similarity of these examples to (2), discussed in Kálmán (2001). The fact that the intuitions described in the latter work concerning the interpretation of the /\-declarative turn out to be very similar to our explanation of (55) and (57) suggests to me that the IBC is on the right track.[20]

Next, (58) illustrates a case where A is unsure about whether her opinion is called for; thus, the metalinguistic issue is whether p addresses the issue on the Table. An analogous example without a taste predicate in (59) exemplifies a different kind of tentativeness, where the speaker is unsure about the speech act itself (i.e., whether the interlocutor is the right person to introduce himself to, that is, whether he is at the right place for his appointment):

(58) B hasn't met A's neighbor, and asks, "What do you think of your new neighbor?" A isn't sure if B wants to know about neighborliness or suitability for dating. A replies:
A: He's attractive? (Malamud & Stephenson 2015, 280, (4c))

(59) (To a receptionist.) Hi, my name is Mark Liberman?[21]
(Malamud & Stephenson 2015, 281, (7))

Although I agree that the assertive rising declaratives in (58)–(59) introduce a metalinguistic issue, I believe that the speaker's commitment to the propositional content of the rising declarative is in both cases actual,

[20] I wish to thank László Kálmán for reminding me of the characterization of the interpretation of (2) in Kálmán (2001), which made me rethink the account in the version of the paper I presented at the birthday workshop.

[21] Original source: Pierrehumbert & Hirschberg (1990, 290).

rather than projected, and does not depend on the hearer's confirmation. ((39), an analogous example, illustrates the problem even better.)

The Hungarian /\-declarative counterparts of these rising declaratives in (58) and (59), shown in (55) above and in (60), respectively, would not be licensed in the same situations:

(60) (To a receptionist.)
#^Engem ^Mark ^Libermannak ^hívnak?
I.ACC Mark Liberman.DAT call.3SG
'My name is Mark Liberman?'

The infelicity of the Hungarian examples is expected on the basis of the IBC: contrary to what clause (a) requires, there is no proposition that the hearer is committed to and the speaker could believe to entail p. Additionally, in the case of (60), the speaker must also previously be committed to the relevant propositional content.

The next example with a vague scalar predicate is an assertive rising declarative again, where "discourse commitments pertain to the appropriate standards of application rather than to objective facts" (Malamud & Stephenson 2015, 281):

(61) A and B are sorting paint cans in a store into a "red" bin and an "orange" bin. B points to orangishred paint and says, "What color would you say this is?" A replies:
A: It's red? (*ibid.*, 281, (8c))

The relevant metalinguistic issue in connection with the above example is whether the standard of redness implicit in p is acceptable, given that A is not confident about her judgment. The corresponding Hungarian example in (62) below is infelicitous in the same situation, as expected, on the basis of clause (a) of the IBC:

(62) #Ez ^piros ^színű?
this red coloured
'It's red?'

The next one is an analogous example, but without a vague scalar predicate:

(63) A teacher (B) is quizzing a student (A) on state capitals. The teacher says: "What's the capital of New York?" The student isn't sure of the answer, but thinks it might be Albany. The student says:
It's Albany? (Malamud & Stephenson 2015, 282, (9c))

As expected, the /\-declarative counterpart of the example above is unacceptable:

(64) A teacher (B) is quizzing a student (A) on state capitals. The teacher says: "What's the capital of New York?" The student isn't sure of the answer, but thinks it might be Albany. The student says:
$^{\#\wedge}$New York állam $^{\wedge}$fővárosa $^{\wedge}$Albany?
New York state capital.its Albany
'The capital of New York state is Albany?'

Again, there is no previous commitment by the addressee that the speaker could be taken to believe that it entails the proposition 'The capital of New York state is Albany', thus clause (a) of the IBC is not satisfied. In the next section we turn to the proposal made in Farkas & Roelofsen (2017).

3.5. Farkas & Roelofsen (2017)

Farkas & Roelofsen (2017, 255) refer to the proposition that corresponds to the surface form of rising declaratives in English (as opposed to its negation) as the *highlighted alternative,* and propose that "both rising declaratives and tag interrogatives signal that the speaker has access to some evidence for the highlighted alternative". In order to account for the compatibility of these two form types with particular contexts, and their incompatibility with others, the authors suggest that the formal representation of discourse contexts proposed by Farkas and Bruce (2010), reviewed in the previous section, should be complemented, for every participant x, with a list referred to as *evidence*(x), which contains the possibilities "for which x has signaled to have some evidence" (*idem.*), and also their "credence level", that is, "the degree to which she believes the alternative itself to be more likely than its complement" (*ibid.*, 20). They suggest that "rising declaratives signal that the speaker's credence in the highlighted alternative α is at most low" (*ibid.*, 256), where low credence means that the speaker considers α to be only somewhat more likely than its negation, $\neg\alpha$.

As an illustration, consider the example in (65):

(65) Student: The answer to this problem is 5 because the square root of 9 is 2 and 2+3 is 5.
Teacher: The square root of 9 is 2? (Farkas & Roelofsen 2017, 269, (55))

I agree with Farkas & Roelofsen (2017) in that (65) "cannot be accounted for in approaches where rising declaratives are taken to signal a 'contingent' or a 'conditional' commitment", as in Gunlogson (2008) and Malamud &

Stephenson (2015). A contingent or conditional commitment means that the speaker is “ready to commit to the highlighted alternative provided that her interlocutor commits to this alternative first” (Farkas & Roelofsen 2017, 270), but in the case of (65) it does not appear to be likely that the teacher would commit to the claim under any circumstances.

The authors’ own explanation sounds as follows. In the case of (65),

> “the available evidence for the highlighted alternative is the student’s prior commitment to it. In this context, the teacher is assumed to be authoritative, that is, she is assumed to know whether the highlighted alternative is true or not. By her use of a rising declarative, she signals to the student that her credence in the highlighted alternative is at most low. Since she is assumed to be authoritative, this can only mean that her credence is zero, and that she is effectively rejecting the student’s prior commitment and urging him to reconsider.” (*op.cit.*, 269)

Although the part of the proposal that derives how the teacher’s use of the rising declarative can be interpreted as rejection of the highlighted alternative is rather resourceful, it remains a question why the teacher chooses the rising declarative form to formulate her question at all, instead of an ordinary positive polar interrogative form, which would unambiguously convey zero credence, and which would also be felicitous.

Furthermore, I think that the proposal by Farkas and Roelofsen (2017) is contradicted by example (66), from Gunlogson (2003).

(66) A: That copier is broken. B’s response:

a. Is it? Thanks, I’ll use a different one.

b. It is? Thanks, I’ll use a different one. (Gunlogson 2003, 21, (45))

In the example above, B is not assumed to have any doubts concerning the truth of the proposition A has committed to. The example in (67) appears similar in many respects to (65), since it also involves an authority asking a question.

(67) Context: Mother sees child putting on cleats:
Mother: What? You are going to play soccer? No way! You are staying home and doing your homework. (Farkas & Roelofsen 2017, 276, (68))

The Hungarian counterpart of (67), equally felicitous in the context, is shown in (68):

(68) ^Indulsz ^focizni?
leave.2SG play.soccer.INF
‘You are going to play soccer?’

(68) can easily be accounted for with the help of the IBC. There is evidence that indicates the truth of the proposition q = 'The child is putting on cleats', which both interlocutors are committed to. If the mother believes that q, independently of any further evidence, entails the truth of p = 'Child is going to play soccer', the /\-declarative is licensed in the context, independently of whether the speaker wants p to become true or not.

Let us finally consider the Hungarian version of (65), which is also felicitous in the relevant context:

(69) A ^kilenc ^gyöke ^kettő?
the nine square.root.its two
'The square root of nine is two?'

I propose that (69) would be accounted for in a manner analogous to echoic uses of /\-declaratives, discussed in Section 3.1. Given the utterance of p = 'The square root of nine is two' by the hearer, the speaker can reasonably be assumed to believe that p is entailed, based on Preparatory Condition 1 of Assertions (Searle 1969, 66), and on the fact that the speaker considers hearer to be observing the Communicative Principle. The acceptability of (69) indicates that its licensing conditions are not influenced by whether the speaker believes in the truth of p or not. The effect of the /\-declarative is, however, influenced by the fact whether the speaker is expected to know whether p or $\neg p$ is the case. If she is considered an expert on the issue, then the fact that she utters the /\-declarative indicates to the hearer that she doubts the truth of the propositional content. If the speaker is not supposed to know whether p is true, her question can simply be taken as asking for confirmation.

The next section summarizes the results of the paper.

4. Conclusions

This paper had two major aims. On the one hand, we investigated the formal properties of the construction type in Hungarian we referred to as the /\-declarative. On the other hand, we looked at existing formal approaches to the felicity conditions of rising declaratives in English, a construction type with an apparently similar distribution, in order to identify the felicity conditions of /\-declaratives in Hungarian and consider possibilities for their formal modelling. In the course of this, we pointed out some aspects of the previous analyses that we considered problematic, but also managed to identify some distributional differences between the two constructions.

We argued that the necessary conditions of the use of the latter should be captured by an account that is a modification of the one proposed by Gunlogson (2003) in a way that assimilates echoic uses of the construction to confirmative or inquisitive uses.

Acknowledgements

I am greatly indebted to László Kálmán for the amount and quality of work he invested in supervising my doctoral dissertation through several years. I can only hope that the present contribution approaches his high standards. I thank an anonymous reviewer and Katalin Mády for critical remarks and suggestions regarding a previous version of this work. The research was supported by the National Research, Development and Innovation Office – NKFIH, under project no. K 115922.

References

Abeillé, A., D. Godard and J.-M. Marandin. 2014. French questioning declaratives in question. In P. Hofmeister and E. Norcliffe (eds.) The core and the periphery. Data-driven perspectives on syntax inspired by Ivan A. Sag. Stanford, CA: CSLI Publications.

Beckman, M. E. and J. B. Pierrehumbert. 1986. Intonational structure in Japanese and English. Phonology Yearbook 3. 255–309.

Beun, R.-J. 2000. Context and form: Declarative or interrogative, that is the question. In H. Bunt and W. Black (eds.) Abduction, belief and context in dialogue: Studies in computational pragmatics. Amsterdam & Philadelphia: John Benjamins. 311–326.

Büring, D. and C. Gunlogson. 2000. Aren't positive and negative polar questions the same? Manuscript, UCLA & UCSC.

É. Kiss, K. 2002. The syntax of Hungarian. Cambridge: Cambridge University Press.

Farkas, D. and K. Bruce. 2010. On reacting to assertions and polar questions. Journal of Semantics 27. 81–118.

Farkas, D. and F. Roelofsen. 2017. Division of labor in the interpretation of declaratives and interrogatives. Journal of Semantics 34. 237–289.

Gärtner, H.-M. and B. Gyuris. 2012. Pragmatic markers in Hungarian: Some introductory remarks. Acta Linguistica Hungarica 59. 387–426.

Ginzburg, J. 1996. Dynamics and the semantics of dialogue. In J. Seligman and D. Westerståhl (eds.) Language, logic and computation. Stanford, CA: CSLI Publications. 221–237.

Ginzburg, J. 2012. The interactive stance. Oxford: Oxford University Press.

Gunlogson, C. 2003. True to form: Rising and falling declaratives as questions in English. New York: Routledge.

Gunlogson, C. 2008. A question of commitment. Belgian Journal of Linguistics 22. 101–136.

Gyuris, B. 2016. A magyar nyelv tagadószós eldöntendő kérdő mondatainak jelentéséhez [On the interpretation of negative polar interrogatives in Hungarian]. Jelentés és Nyelvhasználat 3. 169–190.

Gyuris, B. 2017. New perspectives on bias in polar questions: a study of Hungarian *-e*. International Review of Pragmatics 9. 1–50.

Gyuris, B. 2018. On two types of polar interrogatives in Hungarian and their interaction with inside and outside negation. In C. Dimroth and S. Sudhoff (eds.) The grammatical realization of polarity contrast: Theoretical, empirical, and typological approaches. Amsterdam & Philadelphia: John Benjamins. 173–201.

Hirschberg, J. and G. Ward. 1995. The interpretation of the high-rise question contour in English. Journal of Pragmatics 24. 407–412.

Jeong, S. 2017. Deriving social meanings in an extended Lewisian model. The case of English rising declaratives. Paper presented at the workshop Questioning Speech Acts, Konstanz, 14–16 September 2017.

Jeong, S. 2018. Intonation and sentence type conventions: Two types of rising declaratives. Journal of Semantics 35. 305–356.

Kálmán, L. 2001. Magyar leíró nyelvtan. Mondattan I [Hungarian descriptive grammar. Syntax 1] (Segédkönyvek a nyelvészet tanulmányozásához VI [Supplementary textbooks on linguistics 6]). Budapest: Tinta Könyvkiadó.

Kálmán, L. and Á. Nádasdy. 1994. A hangsúly [Stress]. In F. Kiefer (ed.) Strukturális magyar nyelvtan 2: Fonológia [A structural grammar of Hungarian 2: Phonology]. Budapest: Akadémiai Kiadó. 393–467.

Kenesei, I. 1992. Az alárendelt mondatok szerkezete [The structure of subordinate clauses]. In F. Kiefer (ed.) Strukturális magyar nyelvtan 1. Mondattan [A structural grammar of Hungarian 1. Syntax]. Budapest: Akadémiai Kiadó. 531–713.

Ladd, D. R. 1981. A first look at the semantics and pragmatics of negative questions and tag questions. Proceedings of the Regional Meeting of the Chicago Linguistic Society contents 17. 164–171.

Malamud, S. A. and T. Stephenson. 2015. Three ways to avoid commitments: Declarative force modifiers in the conversational scoreboard. Journal of Semantics 32. 275–311.

Nilsenová, M. 2001. Uncertainty in common ground. ILLC Scientific Publications, MoL-2001-03, University of Amsterdam.

Pierrehumbert, J. and J. Hirschberg. 1990. The meaning of intonation contours in the interpretation of discourse. In P. R. Cohen, J. Morgan and M. E. Pollack (eds.) Plans and intentions in communication and discourse (SDF Benchmark Series in Computational Linguistics). Cambridge, MA: MIT Press. 271–311.

Poschmann, C. 2008. All declarative questions are attributrive? Belgian Journal of Linguistics 22. 247–69.

Reese, B. and N. Asher. 2006. Prosody and the interpretation of tag questions. In E. P. Waldmüller (ed.) Proceedings of Sinn und Bedeutung 11. Barcelona: Universitat Pompeu Fabra. 393–467.

Romero, M. and C.-h. Han. 2004. On negative yes/no questions. Linguistics and Philosophy 27. 609–658.

Rooij, R., van and M. Šafářová. 2013. On polar questions. In R. B. Young and Y. Zhou (eds.) Proceedings of the 13th Semantics and Linguistic Theory (SALT) Conference 2013. Ithaca, NY: Linguistic Society of America and Cornell Linguistics Circle, Cornell University. 292–309.

Šafářová, M. 2007. Nuclear rises in update semantics. In M. Aloni, A. Butler and P. Dekker (eds.) Questions in dynamic semantics. Amsterdam: Elsevier. 295–313.

Searle, J. R. 1969. Speech acts. Cambridge: Cambridge University Press.

Stalnaker, R. C. 1978. Assertion. In P. Cole (ed.) Syntax and semantics 9. New York: Academic Press. 315–332.

Sudo, Y. 2013. Biased polar questions in English and Japanese. In D. Gutzmann and H.-M. Gärtner (eds.) Beyond expressives. Explorations in conventional non-truth-conditional meaning. Leiden: Brill. 277–297.

Szabolcsi, A. 2002. Hungarian disjunctions and positive polarity. In I. Kenesei and P. Siptár (eds.) Approaches to Hungarian 8: Papers from the Budapest conference. Budapest: Akadémiai Kiadó. 217–239.

Varga, L. 2010. Boundary tones and the lack of intermediate phrase in Hungarian (Revisiting the Hungarian calling contour). The Even Yearbook 9. 1–27.

Zeevat, H. 1996. The common ground as a dialogue parameter. In A. Benz and G. Jäger (eds.) Proceedings of MunDial 1997. Munich Workshop on Formal Semantics and Pragmatics of Dialogue. Munich: University of Munich. 195–214.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 281–292

On an unrecognised nonfinite construction in Hungarian

ISTVÁN KENESEI
Research Institute for Linguistics,
Hungarian Academy of Sciences and University of Szeged
kenesei.istvan@nytud.mta.hu

KEYWORDS
complex event nominals
nominalisation
negative concord
negative quantifiers

ABSTRACT
This paper gives a short overview of the debate on a construction in Hungarian whose properties have classified it as nominal or verbal/clausal. On the basis of a new set of data additional criteria from negative quantifiers and negative concord are proposed to differentiate between the relevant characteristics determining its status.

1. Introduction

This article is the latest instalment in a debate on the nature of a particular construction in Hungarian that goes back to about twenty-five years, and even its immediate precursor is nearly ten years old. It began when on the basis of several types of constructions I proposed in Kenesei (2005) that *-ás/és* "nominalisations" or, in Szabolcsi's (1994) terminology, complex event nominals (CENs) behave as nonfinite clauses in a set of well-defined cases.

First, I will summarise my earlier arguments, and then review and argue against the latest development in this exchange, that is, Laczkó's (2009) views to the contrary. Finally, I will put forward new arguments for regarding these constructions as clauses as evidenced by the set of examples to be presented here.

2. Arguments for the clausal analysis of CENs

According to the arguments in Kenesei (2005), the reflexive and reciprocal anaphors in the clauses containing nonfinite verbs, or more precisely, participles, must find their antecedents in their own clauses in (1a) and

(2a), as against the pronominals in (1b) and (2b). "OP" stands for the phonetically empty operator, i.e., relative pronoun, associated with the noun *versek-et* 'poems-ACC'.

(1) a. A lányok$_j$ elolvasták a [[OP fiúk$_i$ által egymáshoz$_i$/magukhoz$_i$
the girls read.3PL the boys by each.other/themselves.ALL
írt] versek-et].
written poems-ACC
'The girls$_j$ have read the poems written by the boys$_i$ to each other$_j$/themselves$_i$.'

b. A lányok$_i$ elolvasták a [[OP fiúk$_j$ által hozzájuk$_i$ írt] versek-et].
the girls read.3PL the boys by they.ALL written poems-ACC
'The girls$_i$ have read the poems written to them$_i$ by the boys$_j$.'

(2) a. A lányok$_i$ elolvasták az [[OP egymáshoz$_i$/magukhoz$_i$ írt] versek-et].
the girls read.3PL the each.other/themselves.ALL written poems-ACC
'The girls$_i$ have read the poems written to each other$_i$/themselves$_i$.'

b. A lányok$_i$ elolvasták a [[OP hozzájuk$_i$ írt] versek-et].
the girls read.3PL the they.ALL written poems-ACC
'The girls$_i$ have read the poems written to them$_i$.'

Without any proper analysis one could say that in (2a) the anaphors are bound by the subject *a lányok* 'the girls', but then (2b) could not be grammatical since in a position where the anaphor can be bound by its antecedent in subject no pronominal can be bound by an antecedent in the same subject position. And since according to the only possible construal of (2b) the poems were written by some person(s) different from the girls, it follows from the Binding Principle that the nonfinite clauses in (2a) and (2b) each contain a covert PRO subject, which compels us to posit the following structures for them, respectively.

(3) a. A lányok$_i$ elolvasták az [[OP PRO$_i$ egymáshoz$_i$/magukhoz$_i$ írt]
the girls read.3PL the each.other/themselves.ALL written
versek-et].
poems-ACC
'The girls$_i$ have read the poems written to each other$_i$/themselves$_i$.'

b. A lányok$_i$ elolvasták a [[OP PRO$_j$ hozzájuk$_i$ írt] versek-et].
the girls read.3PL the they.ALL written poems-ACC
'The girls$_i$ have read the poems written to them$_i$.'

Note that in an ordinary possessive noun phrase both the anaphor and the pronominal can be coreferent with an antecedent outside the NP in question, as with the subjects in (4a–b).

(4) a. A fiúk$_i$ látták [egymás$_i$ rajz-á-t].
the boys saw each-other picture-POSS-ACC
'The boys$_i$ saw each other$_i$'s pictures.'

b. A fiúk$_i$ látták [az ő$_i$ rajz-uk-at].
the boys saw the he picture-POSS.PL-ACC
'The boys$_i$ saw their$_i$ pictures.'

However, in the possessive constructions that contain CENs only anaphors are acceptable. No pronominal coreferent with the subject is tolerated, just as in the examples with nonfinite predicates in (2) and (3).[1]

(5) a. A fiúk$_i$ abbahagyták [egymás$_i$ rajzol-ás-á-t].
the boys stopped.3PL each-other$_i$ draw-DEV-POSS-ACC
'The boys$_i$ stopped drawing each other (lit.: each other's drawing).'

b. *A fiúk$_i$ abbahagyták [az ő$_i$ rajzol-ás-uk-at].
the boys stopped.3PL the he draw-DEV-POSS.PL-ACC
'*The boys$_i$ stopped drawing them$_i$ (lit.: their drawing).'

In view of the consequences of the Binding Principle this scenario is possible only if the bracketed constructions are not DPs but (nonfinite) clauses, whose subjects are empty pronominals, i.e., PROs, coreferential with the respective subjects of the matrix clauses, cf. (6).

(6) a. A fiúk$_i$ abbahagyták [PRO$_i$ egymás$_i$ rajzol-ás-á-t].
'The boys$_i$ stopped PRO$_i$ drawing themselves$_i$.'

b. A fiúk$_i$ abbahagyták [az PRO$_i$ ő$_j$ rajzol-ás-uk-at].
'The boys$_i$ stopped PRO$_i$ drawing them$_j$.'

Kenesei (2005) lists a number of further arguments, including one that is based on antiagreement in case of a third person plural pronominal possessor.[2] It is a well-known feature of Hungarian that whenever the lexical possessor is plural, the possessor DP is marked for plural and its possessum carries a possessive affix unmarked for number, cf. (7a). When,

[1] Obviously, in case the pronominal refers to anyone other than the subject, the construction is grammatical.

[2] As in many other Uralic languages, possessive affixes on the possessed nominal (or possessum) agree with the possessor in all persons if the possessor is a pronominal.

however, a third person plural possessor is a pronominal, its possessum, but not the pronominal, is marked for the plurality of the possessor, cf. (7b).

(7) a. [$_{DP}$ a fiú-k rajz-a/*-uk]
the boy-PL picture-POSS/POSS.3PL
'the boys' picture'

b. az ő rajz-uk
the s/he picture-POSS.3PL
'their picture'

When dative possessors are moved out of their possessive DPs, the possessum has two possible possessive markings available: it could be either unmarked or marked for plural. According to den Dikken's (1999) proposal, when the possessum is unmarked for plural, the possessor has been moved and there is a trace left in its original position, see (8a). When, in turn, the possessum is marked for plural, there is no movement, but the possessor is merged in situ and there is a resumptive pronoun in the possessor position inside the DP, see (8b).

(8) a. A fiúk-nak$_i$ jó volt [a e_i rajz-a]. *unmarked agreement*
the boys-DAT good was the picture-POSS
'The boys' picture was good.'

b. A fiúk-nak$_i$ jó volt [a *pro*$_i$ rajz-uk]. *plural agreement*
the boys-DAT good was the picture-POSS.3PL
'The boys' picture was good.'

I claimed in Kenesei (2005) that CENs did not allow for antiagreement, so following den Dikken (1999), I argued that the possessors/subjects of CENs can only occur in matrix clauses via movement, cf. (9a–b).

(9) a. A problémák-nak$_i$ váratlan volt [a e_i fel-merül-és-e].
the problems-DAT unexpected was the PV-emerge-DEV-POSS
'The emergence of the problems was unexpected.'

b. *A problémák-nak$_i$ váratlan volt [a *pro*$_i$ fel-merül-és-ük].
PV-emerge-DEV-POSS.3PL

Finally, there was also an argument in Kenesei (2005) based on the fact that negation is prohibited in possessive DPs, cf. (10a), while it is always possible in CENs, see (10b). In other words, CENs are propositional, unlike run-of-the-mill possessive DPs.

(10) a. Láttam [a fiúk-nak (*nem) a rajz-á-t].
I-saw the boys-DAT not the picture-POSS-ACC
'I saw the boys' (*not) picture.'

b. Veszélyes volt [a fiúk-nak a le nem rajzol-ás-a].
dangerous was the boys-DAT the PV not draw-DEV-POSS
'(The) not drawing (of) the boys was dangerous.'

3. Arguments against the clausal nature of CENs

In an article written in the framework of Lexical Functional Grammar (LFG) Laczkó (2009) challenges the alleged nonfinite nature of CENs. Although he acknowledges the relevance of the arguments from the Binding Principle, he claims that simpler answers could be found even in a minimalist approach.[3]

As regards my argument from antiagreement, Laczkó is justified in shedding doubt on my judgment of (9b) since he made a limited survey in which some speakers found examples of this type fully acceptable. I must then adjust my position in this regard and suppose that constructions of this kind license resumptive pronouns as was seen in the possessive DP in (8b).

In discussing my third argument, which was based on negation, Laczkó raises a number of problems. One is based on the occurrence of adjectivalising affixes on negated CENs formed from prefixed verbs, cf. (11), where he makes use of the recent borrowing *szével* 'save' in (11a) and the nonsense verb *ki-csaskol* in (11b) to show that their derivatives are not lexicalised.

(11) a. az el nem szével-és-i probléma
the PV not save-DEV-ADJ problem
'the problem of not saving (something on a computer)'

b. a ki nem csaskolás-os jelenség-ek
the PV not *csaskol*-DEV-ADJ phenomenon-PL
'the phenomena of not *kicsaskol*-ing'

Although I have discussed both types of adjectivalisers and shown them to be inflectional affixes, cf. Kenesei (1996; 2014), I did not analyse the

[3] "It is unquestionable that Kenesei's clausal proposal immediately, simultaneously and elegantly solves both the control and the binding problems posed for either Szabolcsi (1994) or Laczkó (1995). Nevertheless, it seems to me that Szabolcsi's account could be modified easily without invoking the whole complex apparatus of clausal syntactic derivation" (Laczkó 2009, 46).

occurrence of the negative operator in the expressions formed with them. The fact that the negative operator is inserted between the preverb and the head bearing the putative nominaliser affix *-ás/és* points to its incompatibility with their derivation within the lexicon. It may be relevant at this point that another construction type formerly categorised as adjectival has also been proved to be a nonfinite clause in Lipták & Kenesei (2017). While I cannot put forward a full analysis here, I suggest that they are extensions of a vP with an implicit internal argument into a NegP, which then undergoes suffixation by *-ás/és*, followed by syntactic affixation by the adjectivalisers *-i* or -V*s*.

Laczkó's claim that the preverb cannot move behind the verb in constructions with CENs as it does in finite clauses, cf. (12a–b) is not a cogent counterargument since the preverb never moves behind the verb in another type of nonfinite, i.e., participial, clause either, cf. (13a–b).

(12) a. A fiú-t nem rajzol-t-am le.
the boy-ACC not draw-PAST-1SG PV
'I didn't draw the boy.'

b. *a fiú nem rajzol-ás-a le
the boy.NOM not draw-DEV-POSS PV
'the not drawing of the boy'

(13) a. a [le nem rajzol-t] fiú
the PV not draw-PPART boy
'the boy not drawn'

b. *a [nem rajzol-t le] fiú
the not draw-PPART PV boy

What is demonstrated here is the well-known property of some Hungarian nonfinite clause types that preserve the original verb final or, in general, head final order of constituents in this Uralic language.

Laczkó has a different objection to the nonfinite analysis based on missing items from the left periphery of finite clauses. The prohibition on *is* 'also, even' and *sem* 'neither, (also/even) not' in (14) and (15) illustrates his point.

(14) a. A fiú-t le is rajzol-t-am.
the boy-ACC PV also draw-PAST-1SG
'I even drew the boy.'

b. *a fiú le is rajzol-ás-a
the boy.NOM PV also draw-DEV-POSS.3SG
'even drawing the boy'

(15) a. A fiú-t le sem rajzol-t-am.
the boy-ACC PV even.not draw-PAST-1SG
'I didn't even draw the boy.'

b. *a fiú le sem rajzol-ás-a
the boy.NOM PV even.not draw-DEV-POSS.3SG
'not even drawing the boy'

There are two options we could follow here in countering Laczkó's point. On the one hand, we could demonstrate that constructions with CENs are projected up to NegP but not beyond, and items from the left periphery such as Topics and quantifiers including phrases headed by *is* and *sem* cannot occur in them – although positive or negative Focus is possible. Focus is below NegP as is illustrated in (16b), where *Anglia* 'England' is interpreted as constituent focus.

(16) a. Sikeres volt [csak Anglia fel-térképez-és-e].
successful was only England PV-chart-DEV-POSS.3SG
'Making a chart only of England was successful.'

b. Tévedés volt [nem Anglia fel-térképez-és-e].
mistake was not England PV-chart-DEV-POSS.3SG
'Making a chart not of England (but of some other country) was a mistake.'

Note that the quantifier phrases headed by *is* or *sem* are impossible even if the preverb is attached to the verb, cf. (17).

(17) a. *Sikeres volt [Anglia is fel-térképez-és-e].
successful was England also PV-chart-DEV-POSS.3SG
Intended meaning: 'Making a chart also of England was successful.'

b. *Tévedés volt [Anglia sem fel-térképez-és-e].
mistake was England neither PV-chart-DEV-POSS.3SG
Intended meaning: 'Making a chart neither of England was a mistake.'

On the other hand, we could claim that the alternative construction with a dative possessor can accommodate quantifier phrases headed by *is* and *sem*, especially since it is generally impossible to insert any independent constituent between the unmarked (or nominative) possessor and the possessed nominal. However, since dative possessors, unlike unmarked ones, can move out of the construction, (18a) is suspect of not being a single constituent, and thereby the dative possessor is understood to be part of a quantifier phrase in the matrix clause, rather than in the embedded one. This is corroborated by (18b), which is ungrammatical since (unlicensed) negation by *sem* cannot follow the matrix predicate, cf. the grammatical (18c) for contrast.

(18) a. Sikeres volt Angliá-nak$_i$ is a [e_i fel-térképez-és-e].
successful was England-DAT also the PV-chart-DEV-POSS
'Making a chart of England was also successful.'

b. *Tévedés volt Angliá-nak$_i$ sem a [e_i fel-térképez-és-e.
mistake was England-DAT neither the PV-chart-DEV-POSS

c. *Nem volt tévedés Angliá-nak$_i$ sem a [e_i fel-térképez-és-e.
not was mistake England-DAT neither the PV-chart-DEV-POSS
'It wasn't a mistake to make a chart of England, either.'

No more of Laczkó's arguments will be discussed here since our purpose was not to enter into a meticulous examination of all arguments for and against but a general overview of the points involved before a new set of phenomena relevant to the issue is presented.

4. New data and analyses

It is well-known that in Hungarian, which is a negative concord language, there has to be a negative operator either c-commanding the negative quantifier in its clause or licensing it by having the negative quantifier in the Spec of the NegP headed by the negative operator.[4]

(19) a. [$_{\text{NegP}}$ Nem [$_{\text{TP}}$ mondott erről senki senki-nek semmi-t (sem)]].
not said-3SG of.this noone noone-DAT nothing-ACC either
'Noone said anything to anyone about this.'

b. [$_{\text{NegP}}$ Senki$_i$ [$_{\text{NegP}}$ senki-nek$_j$ [$_{\text{NegP}}$ semmi-t$_k$ [$_{\text{Neg}}$ nem] [$_{\text{TP}}$ mondott
noone noone-DAT nothing-ACC not said-3SG
e_i e_j e_k erről]]]].
of.this
'Noone said anything to anyone about this.'

[4] See, e.g., Puskás (1998; 2000); Tóth (1999); Surányi (2002; 2006); Olsvay (2006); É. Kiss (2008; 2011); Kenesei (2012).

In case the negative quantifier is in a clause different from the one in which the negative operator occurs, the sentence will be ungrammatical, whether the quantifier was licensed in its root position (20a) or the operator in the matrix clause c-commands it in the embedded clause (20b). For contrast, compare the grammatical (20c) in which the negative quantifier moved from the embedded clause is licensed in the matrix clause.

(20) a. *Anna senki-t$_i$ hisz [hogy e_i nem volt beteg].
Anna noone-ACC believe.3SG that not was sick

b. *Anna nem hiszi [hogy senki volt beteg].
Anna not believe.3SG that noone was sick

c. Anna senki-t$_i$ sem hisz [hogy e_i beteg volt].
Anna noone-ACC not believe.3SG that sick was
'Anna believes noone to have been sick.'

With the exception of infinitives, which behave ambiguously with respect to the independence of their clauses, as seen, e.g., in Szécsényi (2009), the negative operator is required to license the negative quantifier in its clause, whether or not the clause is finite. In (21a) the scope of negation extends to the embedded clause only. In (21b) the negative operator is in the matrix clause, therefore the infinitival clause marked by brackets does not constitute an independent domain.

(21) a. Anna képes volt [PRO semmi-t sem elolvasni].
Anna capable was nothing-ACC not read.INF
'Anna was capable of reading nothing.'

b. Anna nem volt képes [PRO elolvasni semmi-t].
Anna not was capable read.INF nothing-ACC
'Anna wasn't capable of reading anything.'

In contrast to infinitivals, participial clauses are acceptable only if the negative quantifier is licensed by the operator in its own clause.

(22) a. [PRO semmi-t sem észrevéve] Anna belépett a szobá-ba.
nothing-ACC not perceiving Anna entered the room-ILL
'Having perceived nothing, Anna entered the room.'

b. *Anna nem lépett a szobá-ba [PRO semmi-t észrevéve].
Anna not entered the room-ILL nothing-ACC perceiving

(23) a. Anna a [semelyik könyv-et nem olvasó] fiú-t látta.
Anna the none book-ACC not reading boy-ACC saw
'Anna saw the boy reading none of the books.'

b. *Anna a [semelyik könyv-et olvasó] fiú-t nem látta.
Anna the none book-ACC reading boy-ACC not saw

Alberti and Farkas (2017, 828) present a pair of examples relevant to our case, which they do not examine from the aspects discussed here.

(24) a. [$_{\text{DP}}$ Semelyik könyv el-olvas-ás-a] sem okoz gond-ot.
none book PV-read-DEV-POSS not cause problem-ACC
'No problem is caused by reading any of the books.'

b. [$_{\text{NonfinP}}$ Semelyik könyv el nem olvas-ás-a] gond-ot okoz.
none book PV not read-NONFIN-POSS problem-ACC cause
'Reading none of the books causes problems.'

This pair of examples corroborates the position first put forward in Kenesei (2005) based originally on Szabolcsi and Laczkó's (1992) and Szabolcsi's (1994) analyses, namely, that constructions headed by verbs suffixed by *-ás/és* belong to two types: they are either plain (possessive) DPs transparent to negative concord as in (24a), or CENs, that is, nonfinite clauses within whose boundaries the requirements of negative concord are satisfied, as in (24b), where the affix in question is therefore glossed as NONFINITE, rather than DEVERBATIVE, as in (24a). Thus while there is a deverbative derivational affix in (24a), the subject of (24b) is a nonfinite clause. The case of (24a) does not differ from that of an ordinary noun phrase containing a negative quantifier, such as (25).

(25) [$_{\text{NegP}}$ Semelyik könyv [$_{\text{Neg}}$ sem] [$_{\text{TP}}$ okoz gond-ot]].
none book not cause problem-ACC
'None of the books causes problems.'

Since there is a negative operator in the bracketed construction in (24b) that can license the negative quantifier in that very domain and since negative concord works in the minimal clause in this language, this serves as evidence that the CEN in this case is a nonfinite clause.

Acknowledgements

This research was supported by Grant 120073 "Open access book series on the syntax of Hungarian" by the National Office for Research, Development, and Innovation.

References

Alberti, G. and J. Farkas. 2017. Modification. In G. Alberti and T. Laczkó (eds.) Nouns and noun phrases. Comprehensive grammar resources: Hungarian. Amsterdam: Amsterdam University Press. 775–931.

Dikken, M., den. 1999. On the structural representation of possession and agreement: The case of (anti-)agreement in Hungarian possessed nominal phrases. In I. Kenesei (ed.) Crossing boundaries: Advances in the theory of Central and Eastern European languages. Amsterdam & Philadelphia: John Benjamins. 137–178.

É. Kiss, K. 2008. Tagadás vagy egyeztetés? A *senki, semmi* típusú névmások szórendi helye, jelentése és hangsúlyozása [Negation or concord? The word order, interpretation and prosody of *se*-pronouns]. Magyar Nyelv 104. 129–143.

É. Kiss, K. 2011. On a type of counterfactual construction. In T. Laczkó and C. O. Ringen (eds.) Approaches to Hungarian 12: Papers from the 2009 Debrecen conference. Amsterdam & Philadelphia: John Benjamins. 85–107.

Kenesei, I. 1996. A szintagma alapú képzésekről [On syntagm-based derivations]. Nyelvtudományi Közlemények 95. 101–118.

Kenesei, I. 2005. Nonfinite clauses in derived nominals. In C. Piñón and P. Siptár (eds.) Approaches to Hungarian 9: Papers from the Düsseldorf conference. Budapest: Akadémiai Kiadó. 159–186.

Kenesei, I. 2012. Negation in syntactic dialects in Hungarian. In J. Brandtler, D. Håkansson, S. Huber and E. Klingvall (eds.) Discourse & grammar. A Festschrift in honor of Valéria Molnár. Lund: Centre for Languages and Literature, Lund University. 373–386.

Kenesei, I. 2014. On a multifunctional derivational affix: Its use in relational adjectives or nominal modification, and phrasal affixation in Hungarian. Word Structure 7. 214–239.

Laczkó, T. 1995. The syntax of Hungarian noun phrases: A lexical-functional approach. Frankfurt am Main: Peter Lang.

Laczkó, T. 2009. On the *-ás* suffix: Word formation in the syntax? Acta Linguistica Hungarica 56. 23–114.

Lipták, A. and I. Kenesei. 2017. Passive potential affixation: Syntax or lexicon? Acta Linguistica Academica 64. 45–77.

Olsvay, C. 2006. Negative universal quantifiers in Hungarian. Lingua 116. 245–271.

Puskás, G. 1998. On the NEG-criterion in Hungarian. Acta Linguistica Hungarica 45. 167–213.

Puskás, G. 2000. Word order in Hungarian. The syntax of A-bar positions. Amsterdam & Philadelphia: John Benjamins.

Surányi, B. 2002. Multiple operator movements in Hungarian. Doctoral dissertation. LOT, Utrecht.

Surányi, B. 2006. Quantification and focus in negative concord. Lingua 116. 272–313.

Szabolcsi, A. 1994. The noun phrase. In F. Kiefer and K. É. Kiss (eds.) The syntactic structure of Hungarian (Syntax and semantics 27). San Diego/New York: Academic Press. 179–274.

Szabolcsi, A. and T. Laczkó. 1992. A főnévi csoport szerkezete [The structure of the noun phrase]. In F. Kiefer (ed.) Strukturális magyar nyelvtan 1. Mondattan [A structural grammar of Hungarian 1. Syntax]. Budapest: Akadémiai Kiadó. 179–294.

Szécsényi, K. 2009. An LF-driven theory of scrambling in Hungarian infinitival constructions. Doctoral dissertation. SZTE, Szeged.

Tóth, I. 1999. Negative polarity item licensing in Hungarian. Acta Linguistica Hungarica 46. 119–142.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 293–314

On the role of the Speaker's beliefs in some biased questions

ANGELIKA KISS
University of Toronto
angelika.kiss@mail.utoronto.ca

KEYWORDS

rhetorical questions
negative wh-construction
speaker's bias
highlighted proposition
inquisitive semantics

ABSTRACT

This short paper looks at two biased question types, Polar Rhetorical Questions (PRQs) and the so-called Negative Wh-Construction (NWHC). Although at first glance both could seem RQs because of the Speaker's bias toward the negation of the proposition expressed by the sentence radical of the interrogative, it is argued here that the NWHC is not a subtype of RQs. Beliefs attributed to the Addressee play a crucial role in setting NWHCs and PRQs apart.

1. Introduction

The Negative Wh-Construction is a special question type: by its form it is a wh-interrogative, but by its function, it is a denial to some previous utterance.

(1) A: John is a vegetarian.
B: Since whén is John a vegetarian?

Since when in B's reaction bears emphatic stress and expresses that B does not believe the proposition *that John is a vegetarian.* Wh-interrogatives expressing such a move have been observed in a wide range of unrelated languages.

(2) a. Koei bindou jau hai toushugun sik je aa3?! (Cantonese)
he where have be.at library eat thing Q
'No way did he eat anything in the library.'

b. Eti John-i 60 sal i-ni?! (Korean)
where John-NOM 60 year.old be-Q
'No way is John 60 years old.'

c. De dónde va a tener 60 años?! (Spanish)
from where goes to have 60 years
'No way is he 60 years old.' (Cheung 2008, (1a)–(1c))

d. Can ne zaman-dan beri vejetaryan-dır? (Turkish)
can what time-ABL since vegetarian-COP
'Since when is Can a vegetarian?'

e. Mióta érdekel mások véleménye? (Hungarian)
since-when interests others opinion
'Since when do you care about others' opinion?'

Although at a first glance, these questions look similar to RQs, the present paper argues that they are not rhetorical. According to Cheung (2008; 2009), the NWHC has a "negative rhetorical interpretation", by which he means that the at-issue meaning of a Negative Wh-Construction is a negative assertion. There are two discourse-related constraints that apply, both of which are conventional implicatures. One of them he called the "Conflicting View Condition", which restricts the utterance of a NWHC to contexts where the Speaker is sure that her conversational partner believes the opposite of the proposition currently in question, and the other one being the so-called "Mis-Conclusion Condition", which guarantees that NWHCs be used in contexts in which the Speaker thinks that her partner should have come to the same conclusion as the Speaker. The three meaning components are shown in (3), where p is the proposition expressed by the previous utterance to which the NWHC is a denial.

(3) At-issue meaning: $\neg p$
Conventional Implicatures:

a. Conflicting View Condition: The Speaker thinks that the discourse participant believes that p.

b. Mis-Conclusion Condition: The Speaker thinks that the discourse participant should have every reason to believe that $\neg p$. (Cheung 2009, 306)

Applying these to (1), upon uttering 'Since when is John a vegetarian?', the Speaker commits herself to 'John is not a vegetarian', and thinks that her Addressee (wrongly) believes that John *is* a vegetarian. Also, the Speaker thinks that the Addressee should have already known that John was not a vegetarian.

It has been noted that some NWHCs do not necessarily express propositional negation but they can target some metalinguistic issue raised by the latest move (Kiss 2017). The present paper restricts itself to NWHCs that express propositional negation.

1.1. The markedness of NWHCs

NWHCs are thus biased questions, showing a marked form in several ways. As for their syntax, they resist any kind of embedding and the wh-phrase is restricted to a peripheral position. The examples in (4) could only be grammatical if the embedded sentence is pronounced as a quote, but not otherwise. Example (5) shows that a NWHC cannot serve as a sentential subject.

(4) a. *Maria ha detto che ma dove le piaceva il concerto. (Italian)
Maria has said that but where to.her pleasant the concert

b. *Marija skazala chto s kakix eto por chto jej (Russian)
Maria said that from which it time that to.her
ponravilsya koncert.
was.pleasant concert
Intended: 'Maria said that she didn't like the concert.'

(5) a. [Bù chōuyān] yǒuyí jiànkāng. (Mandarin)
not smoke benefits health
'Not smoking is good for your health.'

b. [*Nǎr chōuyān] yǒuyí jiànkāng.
where smoke benefit health
Intended: 'Not smoking is good for your health.'

The peripheral position of the wh-phrase in a NWHC is seen even in wh-in-situ languages. The Cantonese example in (6) shows that unless the wh-phrase is above the ability modal *wui*, the utterance cannot be interpreted as a NWHC. (6a) and (6b) are genuine questions, (6c) is a NWHC.

(6) a. Keoi wui hai bindou maai ce aa3? (Cantonese)
he will at where buy car Q
'Where will he buy a car?'

b. Keoi hai bindou wui maai ce aa3?
he at where will buy car Q
'Where will he buy a car?'

c. Keoi bindou wui maai ce aa3?
he where will buy car Q
Lit.: 'Where will he buy a car?'
'No way will he buy a car' (After Cheung 2008, 23)

Cheung has made the following observations: Languages differ in what wh-phrases can participate in a NWHC, if any, but typically only a small subset of wh-phrases is used, and they cannot be paraphrased. As for their semantics, the wh-phrases do not quantify over the domains canonically associated with them: *where* does not range over places, *when* does not range over times, etc. (Cheung analyzes them as wh-words that stand for possible worlds.) Also, the wh-phrase in a NWHC can easily cooccur with a constituent that in a genuine question would serve as a congruent answer to it.

(7) a. Since when has he been working at UCLA since 2000? (Cheung 2009, (8))
   b. Tā nǎr yǒu zài túshūguǎn lǐ chī fàn (ne)? (Mandarin)
   he where have be.at library in eat meal (Q)
   'No way did he eat anything in the library.'

Another property contributing to markedness is that the wh-phrase in NWHCs can cooccur with predicates having temporal properties that would otherwise make them incompatible in genuine questions. *Since when* is normally ungrammatical with a telic predicate like *decide to vote for Trump*, and *where* is incompatible with stative properties like *be a bus*.

(8) a. Da quando ha deciso di votare per Trump? (Italian)
   since when has decided to vote for Trump
   Lit.: 'Since when did he decide to vote for Trump?'
   b. E bʌs kit^{h}õ ja? (Punjabi)
   this bus where COP
   Lit.: 'Where is this a bus?'

Furthermore, the wh-expressions in NWHCs tend to be marked in form in some languages. They might not be outright idiomatic, but they are somewhere between their ordinary meaning known from genuine questions and the special meaning seen in NWHCs. Examples of such wh-phrases are shown in (9):

(9) a. Italian: *dove* 'where': **(ma) dove* in NWHCs
   b. Cantonese: *(hai) bindou* '(at) where': **(hai) bindou* in NWHCs
   c. Punjabi: *kit^{h}e* 'where': *kit^{h} õ* in NWHCs

English *since when* is close to the literal end of the continuum, but at the other end, we find examples such as *ma dove* 'but where' in Italian and *kit^{h}õ*, derived from *kit^{h}e* in Punjabi.

Lastly, NWHCs bear emphatic stress on the wh-phrase and tend to have a falling final tune, which differentiates them from genuine questions.

The above properties show that NWHCs have a marked form and that the relation of the sentence radical and the wh-phrase is different from how it is in genuine questions. The sentence radical of a NWHC is of the same type as the sentence radical of declaratives and polar questions, that is, they are full propositions. While the sentence radical of wh-interrogatives lacks information (Krifka 2011), the one of NWHCs does not. Examples (7) and (8) prove that the wh-phrase does not come from the proposition as it is in genuine questions. In NWHCs, the wh-phrase seems to act as a discourse-related operator, which is supported by the fact that NWHCs cannot embed.

1.2. What this paper is about

The present paper aims to spell out the difference between RQs and NWHCs by representing the beliefs of the Speaker. This is shown informally in section 2.

In section 3, the system of Farkas & Roelofsen (2017) is presented, which offers a principled way to associate the semantic content and conventions of use of declaratives and polar interrogatives. In this framework, only those utterances can be accommodated that have a proposition-denoting sentence radical and have a so-called highlighted proposition. It will be shown that both PRQs and NWHCs contribute a highlighted proposition. Since this framework also cares about the Speaker's attitude towards the truth of the proposition conveyed by some biased questions, the question whether PRQs and NWHCs should be accommodated in Farkas and Roelofsen's framework becomes relevant.

2. Belief-constellations

2.1. What makes a question rhetorical?

Given Cheung's claim that NWHCs have a 'negative rhetorical interpretation' but are not RQs themselves, it is worth looking at where this interpretation comes from. What makes a question rhetorical? There are at two main views found in the literature on how one could analyze RQs. One of them analyzes them as assertions (Han & Siegel 1997; Han 1998; 2002), and the other, as questions (Rohde 2006; Caponigro & Sprouse 2007).

The analysis that treats them as assertions is based on the fact that RQs pattern with assertions, rather than with questions, in some tests offered by Sadock (1974), some examples of which are shown in (10). A genuine question cannot follow *after all*, but an assertion can. On the other hand, *by any chance* can only appear in a genuine question, not in a RQ or in an assertion. What we see is that RQs pattern with assertions (Cheung 2009).

(10) a. *After all*, do phonemes have anything to do with language?
b. *After all*, phonemes do not have anything to do with language.
c. Does Arthur, *by any chance*, know anything about syntax?

The semantic derivation at some point turns RQs into assertions of the opposite polarity, which is shown in (11). The job is done by an operator that turns the polarity of the sentence radical to the opposite (Han 1998; 2002).

(11) a. Did I tell you that writing a dissertation was easy?
b. *Op*[Did I tell you that writing a dissertation was easy?]
c. ¬[I told you that writing a dissertation was easy] (Han 1998, (25))

Since the polarity operator is set to a pragmatically available value, the expression is not a question any more, it becomes an assertion. One major advantage of the analysis that treats RQs as assertions is that it also accounts for the distribution of Negative Polarity Items in RQs: it is the hidden operator that licenses them.

Caponigro & Sprouse (2007) and Rohde (2006) have pointed out some problems with this analysis, both in terms of empirical facts and theoretically. They noticed that RQs can be answered, while assertions cannot. Also, RQs can have answers that are positive, that is, non-negated propositions.

(12) A: It's understandable that Luca adores Mina. After all, who helped him when he was in trouble?
B: Mina./#Nobody. (Caponigro & Sprouse 2007, (11))

Rohde (2006) points to Schaffer's (2005) *RQs-as-retorts*, another example of RQs not having an answer "of opposite polarity of what has been asked", as Han claims is the case.

(13) a. A: Does Ed McMahon drink?
B: Is the Pope a Catholic? (Schaffer 2005, (5))

b. A: How do you like school?
B: How do you like prison? (*ibid.*, (8))

The RQ in (13a-B) does not convey the negation of what has been asked, on the contrary. And neither does (13b-B), strictly speaking: what it says is 'I like school to the extent you like prison', and the answer also signals that the question was a trivial one. RQs-as-retorts are special in several ways, the reader is referred to Schaffer (2005) for a description of them.

As for the theoretical side of the assertion-like analysis, one cannot overlook the costliness of Han's assumptions. First, the interpretation of interrogatives should branch into questions and assertions, from now on, despite the fact that they can have the same form (at least the same segmental material), as pointed out by Caponigro and Sprouse (2007). Also, Han assumes two sets of wh-words: interrogative and negative ones, yet we have hardly any reason based on morphology to do so.

The other view found in the literature, advocated by Caponigro & Sprouse (2007) and by Rohde (2006), pictures Rhetorical Questions as questions. On this account, it is the obviousness of the answer, and nothing else, that makes a question rhetorical. In other words, genuine information-seeking questions and RQs are the same semantically, although they differ pragmatically; what makes them different is the public beliefs the discourse participants are committed to. Once the speakers all believe the same answer to a given question, it can be asked felicitously as a RQ.

What makes a question rhetorical in Caponigro and Sprouse's analysis is formulated in (14):

(14) a. Q is a Rhetorical Question iff $[\![Q]\!]^w$ is an element of the common ground of the Speaker and the Addressee.

b. Q is an Ordinary Question iff $[\![Q]\!]^w$ is not part of the Speaker's commitments.

The definition of Rhetorical Questions along these lines is given in (15).

(15) A RQ is an interrogative clause whose answer is known to the Speaker and the Addressee, and they both also know that the other knows the answer as well. An answer is not required, but possible. Either the Speaker or the Addressee can answer. (Caponigro & Sprouse 2007, (26))

Treating RQs as questions has some advantages compared to the assertion-like analysis: there is no need to posit two sets of wh-words and the facts about answerability are explained. The major shortcoming of the question-like analysis is that it does not offer an answer to why minimizers can occur in RQs. This problem is not treated here, but the reader is referred to the work of Guerzoni (2004) and Abels (2003) who offer some answer to it. Despite this shortcoming, in this paper, RQs are analyzed as questions to which the answers are perceived by the Speaker as obvious.

2.2. The role of discourse participants' beliefs

A RQ suggesting a negative answer is similar to a NWHC in an important respect: the Speaker believes that the answer to her question is the empty set. What RQs and NWHCs differ in is the belief the Speaker attributes to the Addressee about the truth of the proposition in question. Following Caponigro and Sprouse, in the case of a RQ, the Speaker attributes a positive or negative belief to herself about the proposition in question p if the RQ in question is a polar one. If the RQ is a wh-interrogative, this negative belief means that according to the Speaker, there is no value that makes the sentence radical a true proposition. At the same time, the Speaker attributes the same belief on the proposition p to her Addressee, which is either the sentence radical of a PRQ or a proposition that serves as a true answer to the wh-RQ. This condition holds whether it is a RQ awaiting a positive or a negative answer.

In the case of a NWHC, recall that there are some discourse-related constraints, namely the Conflicting View Condition and the Mis-Conclusion Condition. These guarantee that a NWHC is pronounced felicitously only if the Speaker and the Addressee have opposing beliefs about p. Upon reacting to the previous utterance expressing the proposition p, a NWHC expresses $\neg p$.

Let S^S represent the Speaker's own belief about the proposition in question, and let S^A be the belief attributed by the Speaker to the Hearer. The polarity of the belief about the truth of p is marked by $+$, $-$ and *neut* for positive, negative and neutral stances on it. In this case, RQs with negative answers, RQs with positive answers and NWHCs pertain to three different constellations of beliefs. Let us represent a certain belief-constellation as an ordered pair of Speaker's beliefs, the first element being S's belief about p and the second, what S attributes to A as a belief about p.

(16) a. Rhetorical Question with a negative answer: $\langle S^{S-}, S^{A-}\rangle$
b. Rhetorical Question with a positive answer: $\langle S^{S+}, S^{A+}\rangle$
c. Negative Wh-Construction: $\langle S^{S-}, S^{A+}\rangle$

Both RQs and NWHCs are biased questions, since in both cases, the Speaker has a bias toward the proposition she is questioning, and in addition, she also attributes a bias to the Addressee. The similarity between RQs suggesting a negative answer and NWHCs is due to the Speaker's negative bias S^{S-} in both cases. But the Addressee is attributed different things in the two cases: RQs with a negative answer require the Addressee's belief to be negative as well, while NWHCs are felicitous only if the Addressee's belief on the subject matter p is of opposite polarity, i.e., positive.

These properties are summarized in the following table, where S stands for Speaker, and A for the belief attributed to the Addressee by S:

Table 1: Speaker's beliefs about the proposition in question and about Addressee's beliefs on it

S's belief about A $\rightarrow$ S's own belief	S^{A-}	$S^{A.neut}$	S^{A+}
S^{S-}	**RQ**$^{-}$	accommodate as a RQ	**NWHC**
$S^{S.neut}$	n/a	genuine question	rising declarative
S^{S+}	n/a	accommodate as a RQ	**RQ**$^{+}$

The two dimensions of the table show the possible beliefs of the Speaker about the truth of the proposition p. The header column shows that the Speaker can believe that p is not true (S^{S-}), that p is true (S^{S+}), or she can have a neutral stance on it ($S^{S.neut}$). The header row shows three possible beliefs the Speaker can attribute to the Addressee: she can think A agrees with p (S^{A+}), that A disagrees with p (S^{A-}), or it is possible that the Speaker has no reason to attribute any bias to A ($S^{A.neut}$). Attributing a neutral stance to A does not mean that A does not know whether p is true, it only means that the Speaker is unable to attribute a positive or a negative stance to A about p, which does not exclude that A does have one. $S^{S.neut}$, on the other hand, should be read off as "S cannot attribute either a positive, or a negative attitude to S", that is, to herself, which means that the Speaker does not know whether p is true or not.

(17) Belief-constellations

Any Speaker S in a conversation who pronounces an interrogative with a sentence radical p has a belief constellation represented as an ordered pair $\langle S^S, S^A \rangle$,
where S^S indicates the belief about the truth, falsity or a neutral stance of p attributed by S to herself, and
where S^A indicates the belief of the truth, falsity or neither, of p, attributed by S to the Addressee A,
so that both S^S and S^A can have positive, negative or neutral values, depending on the type of the question.

If we adopt Caponigro and Sprouse's view on the "felicity conditions" of RQs, then both members of the ordered pair should show the same (non-neutral) value. To illustrate it, consider the following example of a RQ awaiting a negative answer.

(18) Do chickens have lips? (Schaffer 2005, (1))

Here, the proposition in question p would be the sentence radical of the RQ, 'chickens have lips'. The answer to this question is negative and this fact is obvious enough so that we can rely on that the Addressee also knows this. The belief constellation of the Speaker of (18) is therefore $\langle S^{S-}, S^{A-} \rangle$.

(19) A: It's understandable that Luca adores Mina. After all, who helped him when he was in trouble?
A/B: Mina./$^{\#}$Nobody. (Caponigro & Sprouse 2007, (11))

The proposition in question here is composed of applying the commonly known answer to the question radical (in the terminology of Krifka 2017): $\lambda x[x$ helped Luca when he was in trouble$]$(Mina). The proposition in question in this case would be 'Mina helped Luca when he was in trouble'. The Speaker believes this proposition to be true and expects her Addressee to also believe this to be the case, the belief constellation is thus $\langle S^{S+}, S^{A+} \rangle$.[1]

As for the rest of the cases in Table 1, a Speaker can use a RQ felicitously even if she does not believe the Addressee to be biased towards the answer her question suggests because as Abels (2003) noted, a cooperative discourse participant can accommodate presuppositions arising from RQs. This represents the notion of "one-sided mutual belief", the importance of which is highlighted by Pierrehumbert and Hirschberg (1990). Belief constellations that have a neutral value for S's own belief about p are genuine

[1] The belief-constellation $\langle S^{S+}, S^{A+} \rangle$ is somewhat underspecified in this case. It only shows that there is a true answer to the RQ, but the exact information, that is, which element of the denotation of the wh-question is true, comes from world knowledge.

questions in case the belief attributed to A is not biased either: $\langle S^{S.neut}, S^{A.neut}\rangle$. If A is biased towards the positive answer, however, it is a case of a rising declarative (Gunlogson 2001). $\langle S^{S.neut}, S^{A+}\rangle$ represents cases of questions that in English are syntactically declarative but have a rising intonation. Rising declaratives fail to commit the Speaker to the truth of the sentence radical, yet they are biased, namely their Speaker attributes a certain proposition to the Addressee (*ibid.*, 4).

In sum, Rhetorical Questions and the Negative Wh-Construction are similar in an important respect, namely that the Speaker's negative stance on a proposition in question p (S^{S-}) is part of the felicity conditions of both. What sets them apart is the belief the Speaker attributes to the Addressee.

3. Semantic content and conventions of use

Farkas and Roelofsen (2017) suggest a framework that allows that semantic content and conventional discourse effects divide the labor when interpreting declaratives and polar interrogatives. Besides information-seeking questions, they also look at some special questions such as rising declaratives, claiming that their framework can host all question types, biased and neutral. It is argued here that we have reasons to accommodate RQs and NWHCs in their system, and if so, some extensions are needed.

The authors claim that the difference in the interpretation of assertions and polar interrogatives follows from their semantic differences while the same convention of use is assigned to both. Marked utterance types, for example rising declaratives, are also subject to some special discourse effects, which is due to how strongly the Speaker believes that the highlighted proposition (the sentence radical) is true. This they call the *credence level*, and it becomes important whenever special effects arise, and special effects arise due to markedness.

(20) Division of labor principle (Farkas & Roelofsen 2017, 250)

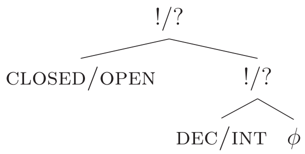
a. The discourse effects of unmarked forms should be fully determined by their semantic content and the basic convention of use, F_b.

b. The discourse effects of marked forms should always include the discourse effects that are dictated by their semantic content and the basic convention of use F_b. In addition, they may include special discourse effects connected to the particular sentence type involved.

The authors characterize the basic discourse context in terms of the set of discourse participants, the table, which is a stack of propositions that have been raised as issues but have not yet been accepted into the common ground (Farkas & Bruce 2010), and the set of commitments, which maps discourse participants to the set of propositions that cover all their public beliefs (to their commitment set, see Stalnaker (1978)).

Farkas and Roelofsen characterize the semantics of declaratives and interrogatives in terms of clause type markers. The presence of the clause type markers DEC or INT are signaled in English by word order, and the clause type markers CLOSED and OPEN are signaled by falling and rising intonation, respectively. The two kinds of markers together determine whether the sentence is inquisitive or not (Ciardelli et al. 2013). The sentence radical is the argument of the DEC/INT marker, and the resulting expression then combines with the CLOSED/OPEN marker, which can have a further effect on its inquisitiveness.

(21) Clause type markers



(22) a. $[\![\text{DEC}]\!] = \lambda P.!P$

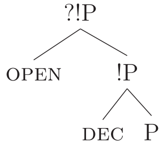
b. $[\![\text{INT}]\!] = \lambda P.\langle ? \rangle$

c. $[\![\text{CLOSED}]\!] = \lambda P.P$

d. $[\![\text{OPEN}]\!] = \lambda P.?P$

In the case of an assertion, the clause type marker DEC makes any sentence radical non-inquisitive (marked by !), and if it is also CLOSED, the utterance ends up non-inquisitive, that is, it is informative and does not need further information to be completed. On the other hand, polar interrogatives expressing an information-seeking question are typed INT and OPEN, marked by ?. Marked sentence types typically have both ! and ?, as it is exemplified by the case of rising declaratives.

(23) Clause type markers in rising declaratives



Since in this case a declarative combines with OPEN, both the highlighted alternative P and its complement are added to the table, and the sentence becomes inquisitive. A proposition is equivalent to a set of worlds: adding the proposition together with its complement results in adding the whole set of worlds W to the table, which is not an informative move.

The second interpretational component is determined by the convention of use.

(24) Basic convention of use
If a discourse participant x utters a declarative or interrogative sentence ϕ, the discourse context is affected as follows:

a. The proposition expressed by ϕ, $[\![\phi]\!]$, is added to the **table**.

b. The informative content of ϕ, $\bigcup[\![\phi]\!]$, is added to the **commitments**(x).

In case of an assertion, ϕ equals to a proposition, but in case of a polar interrogative, ϕ stands for both the highlighted alternative and its complement, as shown in (25). Thus, in the latter case, the entire set of worlds W is added to **commitments**(x), as it becomes an inquisitive proposition, resulting in a trivial commitment, requiring an answer.

(25) Conventional discourse effects of participant x uttering a polar interrogative expressing the proposition $\{\alpha, \overline{\alpha}\}$:

a. $\{\alpha, \overline{\alpha}\}$ is added to the **table**

b. W is added to the **commitments**(x). (Farkas & Roelofsen 2017, 267)

In case of a genuine question, the answer is supposed to reduce the set of worlds in **commitments**(x). But in marked sentences, some special discourse effects may arise. Between any two forms that have the same semantic content, the one that is formally more complex, and therefore "less likely to ensure communicative success" is considered marked (*ibid.*, 263). Assuming that both segmental and suprasegmental material counts as 'form', biased questions are all marked, as expected. Tag interrogatives are longer than polar interrogatives without a tag, and rising declaratives, RQs and NWHCs are marked because of their intonation. And NWHCs, in addition, also have peculiar syntactic properties that in their genuine counterparts would cause ungrammaticality.

Marked sentences are subject to the same basic convention as unmarked ones, but in addition, they carry some extra information that is signaled by their non-minimal form. The extra information, according to Farkas and Roelofsen, concerns the level of credence the Speaker has in the truth of the highlighted proposition. In characterizing the discourse context, the authors make use of a set they called **evidence**(x), which stands for the set of propositions such that speaker x has access to evidence to believe them (*evidenced possibilities* as Farkas and Roelofsen call it). The Speaker, in addition to signalling that she has evidence to the truth of the highlighted proposition, also signals the level of confidence she has in the truth of it. She can have high credence, as in the case of uttering an assertion, or low credence, when uttering a rising declarative, or in yet other cases, moderate or zero credence. The set **evidence** is thus made up of ordered pairs $\langle p, i \rangle$ such that p is a proposition the Speaker has evidence to believe and i is an interval of credence levels. In the case of a rising declarative, the special effect equals to a low level of credence on the part of the Speaker.

(26) Special conventional discourse effects of a rising declarative:

When a discourse participant x utters a rising declarative ϕ, expressing the proposition $[\![\phi]\!]= \{\alpha, \overline{\alpha}\}$, the discourse context is affected as follows:

1. Basic effects: as in (25)
2. Special effect: $\langle \alpha, [zero, low] \rangle$ is added to **evidence**(x).

(Farkas & Roelofsen 2017, 269)

Farkas and Roelofsen's system thus determines whether a sentence is inquisitive or not by using clause type markers. These sentences are then subject to the same basic convention, namely that the inquisitive or non-inquisitive expression is added to the table and the corresponding set of worlds is added to the set of commitments. Thirdly, if the sentence is marked, some special effects are postulated, which share one characteristic: they all say something about the level of the Speaker's confidence in the truth of the highlighted proposition. However, the authors admit that special effects are assigned to each sentence form individually, but not according to a principle.

3.1. PRQs and the NWHC in Farkas and Roelofsen's system

Farkas and Roelofsen do not consider the case of RQs or NWHCs. They mention that RQs are used to underline some proposition already present in the common ground, but this property of them belongs to pragmatics. However, both RQs and NWHCs are marked utterances based on the author's measures, because of the mismatch between their form (an interrogative) and what they convey (an assertion), which makes them "more prone to misinterpretation". Intonation also alters the form so that it becomes more marked, as RQs are more likely to have a sentence-final falling intonation (Banuazizi & Creswell 1999), and NWHCs have a similar tendency.

There are some other good reasons to treat PRQs and NWHCs in the same system, as they both have sentence radicals that are full propositions. This is obvious for PRQs, but less so for NWHCs. In section 1.1, it was shown that the wh-expression cannot stand for a constituent that comes from the clause, thus NWHCs are made up of a wh-phrase independent of the clause (a fact also observed by Cheung (2009)), and a proposition-denoting sentence radical.

Also, these sentence radicals pass the tests of highlighted propositions suggested by Farkas and Roelofsen, namely they can serve as antecedents for anaphoric expressions. Such tests include expressions like *yes*, *no*, *if so* and *otherwise*. Just as the sentence radicals of genuine polar questions, rising declaratives, and tag questions can be referred to by such expressions, as shown in (27), so can the sentence radicals of NWHCs (28b) and PRQs (29c), too, setting them apart from wh-interrogatives, whether they are genuine or rhetorical (28a).

(27) a. János vegetáriánus? (genuine)
John vegetarian
'Is John a vegetarian?'

b. János/\ vegetáriánus/\? (rising declarative)[2]
'John's a vegetarian?'

c. János vegetáriánus, (vagy) nem? (reverse polarity tag question)
John vegetarian or no
'John is a vegetarian, isn't he?'

d. Igen./Így van.
yes/so is

(28) a. A: János mióta vegetáriánus? (genuine)
John since.when vegetarian?
'Since when is John a vegetarian?'

B: #Igen./#Így van.
yes/so is

[2] Hungarian polar questions with multiple rise-fall (indicated by Kálmán 2001 as /\) and English rising declaratives do not have the exact same distribution, although there is an overlap, see Gyuris (2019).

b. János mióta vegetáriánus? Ha így lenne, halat se ehetne. (NWHC)
John since.when vegetarian if so were fish either could.eat
'Since whén is John a vegetarian? If it were the case, he couldn't eat fish either.'

c. János talán vegetáriánus? Mert ha nem így van,
John perhaps vegetarian because if not so is
akkor ne prédikáljon állatkínzásról. (PRQ)
then not preach.SUBJ about.cruelty.to.animals
'Is John a vegetarian? Because if not, he should not preach about cruelty to animals.'

The same contrast is shown in English with (29): let alone genuine wh-interrogatives, all other question types under discussion have highlighted propositions that *it* can refer back to.

(29) a. A: Is John a vegetarian? (genuine)
B: I don't believe it.

b. A: John's a vegetarian? (rising declarative)
B: I don't believe it.

c. A: John is a vegetarian, isn't he? (reverse polarity tag question)
B: I don't believe it.

d. A: Since when is John a vegetarian? (genuine)
B: #I don't believe it.

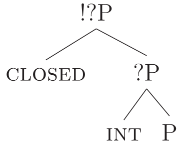
e. A: Since whén is John a vegetarian? (NWHC)
B: I don't believe it either.

f. A: Is John a vegetarian? (PRQ)
B: I don't believe it either.

Given that the PRQs and NWHCs seem to contribute highlighted propositions to the discourse just like assertions, polar interrogatives, rising declaratives and tag questions, there is reason to accommodate them in the same framework.

If INT contributes inquisitiveness and CLOSED encodes falling intonation, a PRQ or a NWHC should then have the following clause type markers:

(30) Clause type markers of PRQs and NWHCs



In the case of a PRQ, the argument of INT is a proposition, which needs INT to become inquisitive. Since CLOSED is semantically vacuous (see (22c)), the result is an inquisitive expression, as expected. The interpretation is shown in (31b).

(31) a. (Come on,) Is John a vegetarian?
  b. ⟦INT John is a vegetarian⟧ = {{John is a vegetarian}, {John is not a vegetarian}}

The conventional discourse effects of a PRQ suggesting a negative answer are the same as with a genuine polar question. But because of its markedness, it is expected to have special effects. If the highlighted alternative is the sentence radical, the special effect should be a strong disbelief in it, given its typically falling final tune. This, according to Farkas and Roelofsen (2017, 256), is captured by a *zero* credence level, which can mean that the Speaker is unbiased or that she is against the truth of the highlighted proposition.

(32) Special conventional discourse effects of a Polar Rhetorical Question awaiting a negative answer (first version):

  Basic effects:

  a. $\{\alpha, \overline{\alpha}\}$ is added to the **table**

  b. W is added to the **commitments**(x).

  Special effect: $\langle \alpha, [zero] \rangle$ is added to **evidence**(x).

The problem that arises here is that the basic and special conventional discourse effects of a PRQ awaiting a negative answer is not any different from a genuine polar question. Since in the case of a PRQ awaiting a negative answer, the Speaker believes the opposite of the highlighted proposition, saying that she has a *zero* credence level is not enough. It is suggested that the label *contrary* be used in this case, and let *zero* be saved for unbiased questions.

(33) Distinguishing genuine and biased polar questions by their special effects (final version):

  a. Special effect of a genuine polar question:
  ⟨*John is a vegetarian*, [*zero*]⟩ is added to **evidence**(x).

  b. Special effect of a PRQ with a negative answer:
  ⟨*John is a vegetarian*, [*contrary*]⟩ is added to **evidence**(x).

Now, the case of NWHCs is peculiar and it is well possible that they cannot be accommodated in Farkas and Roelofsen's system at all. In what follows, we give a first look at the possibilities we have.

A genuine wh-interrogative is expected to put a Hamblin-set on the table.

(34) a. Since when is John a vegetarian? (genuine question)

b. ⟦INT John is a vegetarian since t ∈ TIMES⟧ = {{w: John is a vegetarian since t_1 ∈ TIMES}, {w: John is a vegetarian since t_2 ∈ TIMES}, ..., {w: John is a vegetarian since t_n ∈ TIMES}}

Recall that NWHCs differ from ordinary wh-interrogatives because they have sentence radicals denoting full propositions unlike genuine wh-interrogatives which have question radicals (Krifka 2017); and because they contribute a highlighted proposition, unlike genuine wh-interrogatives. Also, the wh-phrase in a NWHC does not range over entities it is canonically associated with, e.g. *since when* in (1B) does not range over times (Cheung 2008; 2009), which is corroborated by the fact that the segmental material of wh-phrases in NWHCs is slightly altered in some languages. If (34a) is pronounced as a NWHC, we cannot use the variable t standing for time-points to create the partition on W, because the wh-phrase is no longer associated with its canonical domain, as it was shown in (7), and the same applies to NWHCs containing other wh-phrases.

Otherwise, we can say that the highlighted proposition is the sentence radical of the NWHC and put it on the table along with its complement generated by INT, as a result of which we get the following basic conventional discourse effects:

(35) Basic conventional discourse effects of a Negative Wh-Construction (provisional):

a. {α, $\overline{\alpha}$} is added to the **table**

b. W is added to the **commitments**(x).

But (35) is not a fortunate representation either, because we represent a wh-interrogative (though not a canonical one) the same way we represent a polar interrogative. Until a better way of representing the conventional discourse effects of NWHCs is found, we let (35) be the provisional representation.

3.2. Special effects

As for the special effects of NWHCs, the Speaker is expected to believe that the highlighted proposition is false, that is, she will be assigned the credence level *contrary*, as in the case of PRQs awaiting a negative answer.

(36) Special effect of a NWHC: $\langle \alpha, [contrary] \rangle$ is added to $\mathbf{evidence}(x)$.

If PRQs awaiting a negative answer and NWHCs have the same basic and special conventional discourse effects, cf. (32), (33), (35) and (36), then they need to be further distinguished. It was suggested in section 2 that RQs awaiting a negative answer and NWHCs both share the property that their Speaker disagrees with some proposition *p*, possibly conveyed by the latest move, which is also the sentence radical of the sentence. It was also shown that NWHCs are not RQs, because the Speaker attributes different beliefs to her Addressee in both cases. Namely, in the case of RQs, the Speaker believes and/or conveys that her Addressee has (or should have) the same belief on the truth of *p*, while in the case of NWHCs, the Speaker and Addressee have opposing beliefs. If so, a more adequate description of special effects would be one that includes an extra slot for the belief attributed to the Addressee. NWHCs, PRQs and Rising Declaratives would then be described as follows:

(37) a. Special effect of a NWHC: $\langle S\langle \alpha, contrary \rangle, A\langle \alpha, high \rangle \rangle$

b. Special effect of a PRQ suggesting a negative answer: $\langle S\langle \alpha, contrary \rangle, A\langle \alpha, contrary \rangle \rangle$

c. Special effect of a rising declarative $\langle S\langle \alpha, [zero, low] \rangle, A\langle \alpha, high \rangle \rangle$

Although representing both the Speaker's belief and the belief attributed by the Speaker to the Addressee is more costly, as it complicates the representation, there seems to be no other way to capture the difference between NWHCs and PRQs with a negative answer, at least given the premises this paper builds on.

There are various views in the literature on how it comes about that RQs convey their speakers' bias: it has been attributed to semantics (Han 2002) and to pragmatics (Caponigro & Sprouse 2007). As for NWHCs, Cheung argues that the special effect described here, which he called the Conflicting View Condition, is a conventional implicature.[3] His claim is not inconsistent with the arguments of this paper. What we stated here is that PRQs and NWHCs, being marked utterances, have a highlighted proposition, based on the division of labor principle in (20), therefore they must have special effects. These special effects are present by virtue of the marked form, which means they are encoded, and being encoded is one of the tests Potts (2015) uses to tell conventional implicatures from other

[3] At least he has shown that the Conflicting View Condition associated with NWHCs is not at-issue and not cancellable either.

kinds of meaning. There are other tests though, and it is outside the scope of this paper to determine where these special effects belong, although it is not a minor issue.

Farkas and Roelofsen (2017, 272) suggest that intonation and credence levels show a connection in marked sentence forms. The generalization they make is the following: the lower the credence level in the highlighted alternative, the less likely the utterance ends with a falling tune; and the higher the credence level, the less likely the utterance involves a final rise. If NWHCs have the special discourse effect shown in (36), in favor of which some supporting facts have been shown, this generalization does not hold. In (1B), because of the falling final tune, the Speaker should strongly believe the truth of the sentence radical 'John is a vegetarian', which is just the opposite of how it actually is. Thus, as a next step, this prediction should be modified to possibly accommodate the belief attributed by the Speaker to the Addressee as well.

Lastly, a remark should be made about the status of RQs. What was considered here as special effect, Farkas and Roelofsen considers pragmatics. Undoubtedly, not all PRQs are marked the way suggested here. Banuazizi and Creswell (1999) observed a tendency for RQs to have a sentence-final falling intonation, but by no means is this always the case for (English) PRQs. What is more, as Farkas and Roelofsen claim, genuine polar interrogatives can have a sentence-final fall, too. If so, it raises the question of how marked RQs are at all, and in what ways.

4. Conclusion

This short paper has presented some arguments for the importance of belief-constellations, that is, for representing a more elaborate picture of the Speaker's own beliefs and those attributed to others. The advantage of doing so is seen in a commitment-based account such as the one of Farkas & Roelofsen (2017), that aims to host sentences having a highlighted alternative. PRQs with a negative answer and NWHCs both have such alternatives, so there is reason to consider their place in such a model. Fitting them into Farkas and Roelofsen's account raises questions; to some of them, the paper suggested an answer.

Acknowledgements

I am thankful to Michela Ippolito and Guillaume Thomas for the insights they gave while I was working on the Negative Wh-Construction earlier. I am also thankful for Beáta Gyuris and Lisa Cheng for having sparked my interest in the semantics of questions. But most importantly, I am thankful for László Kálmán, for having sparked my interest in semantics itself.

References

Abels, K. 2003. Who gives a damn about minimizers in questions? In R. Young and Y. Zhou (eds.) Proceedings of Semantics and Linguistic Theory (SALT) 13. Ithaca, NY: Linguistic Society of America and Cornell Linguistics Circle, Cornell University. 1–18.

Banuazizi, A. and C. Creswell. 1999. Is that a real question? Final rises, final falls and discourse function in yes-no question intonation. In CLS 35. 1–14.

Caponigro, I. and J. Sprouse. 2007. Rhetorical questions as questions. In E. Puig-Waldmüller (ed.) Proceedings of Sinn und Bedeutung. Barcelona: Universitat Pompeu Fabra, 121–133.

Cheung, L. Y.-L. 2008. The Negative Wh-Construction. Doctoral dissertation. UCLA.

Cheung, L. Y.-L. 2009. Negative *wh*-construction and its semantic properties. Journal of East Asian Linguistics 18. 297–321.

Ciardelli, I., J. Groenendijk and F. Roelofsen. 2013. Inquisitive semantics: a new notion of meaning. Language and Linguistic Compass 7. 459–476.

Farkas, D. and K. B. Bruce. 2010. On reacting to assertions and polar questions. Journal of Semantics 27. 81–118.

Farkas, D. and F. Roelofsen. 2017. Division of labor in the interpretation of declaratives and interrogatives. Journal of Semantics 34. 237–289.

Guerzoni, E. 2004. *Even*-NPIs in yes/no questions. Natural Language Semantics 12. 319–343.

Gunlogson, C. 2001. True to form: Rising and falling declaratives as questions in English. Doctoral dissertation. University of California Santa Cruz.

Gyuris, B. 2019. Thoughts on the semantics and pragmatics of rising declaratives in English and rise-fall declaratives in Hungarian. Present volume.

Han, C.-H. 1998. Deriving the interpretation of rhetorical questions. In Proceedings of WCCFL 16. Stanford: CSLI Publications, 237–253.

Han, C.-H. 2002. Interpreting interrogatives as rhetorical questions. Lingua 112. 201–229.

Han, C.-H. and L. Siegel. 1997. Syntactic and semantic conditions on NPI licensing in questions. In Proceedings of WCCFL 15. Stanford: CSLI Publications, 177–191.

Kálmán, L. 2001. Magyar leíró nyelvtan. Mondattan I [Hungarian descriptive grammar. Syntax 1] (Segédkönyvek a nyelvészet tanulmányozásához VI [Supplementary textbooks on linguistics 6]). Budapest: Tinta Könyvkiadó.

Kiss, A. 2017. Meta-conversational *since when*-questions and the common ground. In V. Petukhova and Y. Tian (eds.) Proceedings of the 21st Workshop on the Semantics and Pragmatics of Dialogue. Saarbrücken, 38–47.

Krifka, M. 2011. Questions. In C. Maienborn, K. von Heusinger and P. Portner (eds.) Semantics: An international handbook of natural language meaning. Volume 1. Walter de Gruyter: Berlin & Boston. 1742–1785.

Krifka, M. 2017. Negated polarity questions as denegations of assertions. In C. Lee, F. Kiefer and M. Krifka (eds.) Contrastiveness in information structure, alternatives and scalar implicatures. Cham: Springer. 359–398.

Pierrehumbert, J. and J. Hirschberg. 1990. The meaning of intonation contours in the interpretation of discourse. In P. R. Cohen, J. Morgan and M. E. Pollack (eds.) Plans and intentions in communication and discourse (SDF Benchmark Series in Computational Linguistics). Cambridge, MA: MIT Press. 271–311.

Potts, C. 2015. Presupposition and implicature. In S. Lappin and C. Fox (eds.) The handbook of contemporary semantic theory (Second edition). Cambridge, MA & Oxford: Wiley Blackwell. 168–202.

Rohde, H. 2006. Rhetorical questions as redundant interrogatives. San Diego Linguistics Papers 2. 134–168.

Sadock, J. 1974. Towards a linguistic theory of speech acts. New York: Academic Press.

Schaffer, D. 2005. Can rhetorical questions function as retorts? Is the pope catholic? Journal of Pragmatics 37. 433–460.

Stalnaker, R. C. 1978. Assertion. In P. Cole (ed.) Syntax and semantics 9. New York: Academic Press. 315–332.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 315–352

Indexing old literary Finnish text

KIMMO KOSKENNIEMI
University of Helsinki
kimmo.koskenniemi@helsinki.fi

PIRKKO KUUTTI
Institute for the Languages of Finland
pirkko.kuutti@kotus.fi

KEYWORDS

two-level morphology
Old Literary Finnish
historical texts
finite-state transducers
HFST

ABSTRACT

A purpose of this study was to test the Helsinki Finite-State Transducer (HFST) technology tools, including its hfst-twolc compiler, the use of weighted finite-state transducers, to use the HFST tools out of Python scripts, and to use them together for comparing two related language forms. A strict procedure was followed in constructing, testing and revising two-level rules which relate written Modern Standard Finnish and Old Literary Finnish as used in the 17th century Bible. In particular, the advantages of the strict independence of the two-level rules were utilised. No practical production system was planned, but the results could be quite useful for indexing and concordancing similar Old Literary Finnish texts.

1. Corpus

A corpus of readily available old Finnish texts was needed for the study, more specifically texts whose language was sufficiently different from Modern Standard Finnish (MSF), but where the variation within the corpus was reasonable. The Finnish language used between the years 1540 and 1820 which is called Old Literary Finnish (OLF, in Finnish *vanha kirjasuomi*) is sufficiently distinct from MSF for the purposes of this study. Morphological analysers for MSF cannot be used as such for any OLF texts. The differences are greater the further one goes back in time.

The Finnish translation of the Bible from 1642 (often called *Biblia*) seemed suitable for the purposes of this project. Its language is homogenous enough and the text of Biblia is available as a digital text from the

Kaino[1] service of The Centre for Languages in Finland (Kotus). The whole translation consists of some 900,000 word tokens. For the present study, the fourth part of the Old Testament (VT-4, some 20,000 word tokens) and the first part from the New Testament (UT-1, some 12,000 word tokens) were selected and used together as our corpus.[2] A smaller corpus could have been sufficient for the design of the rules, but one needed a fair amount of text in order to extract a list of common word forms.

The material chosen was fairly but not fully homogeneous. Orthographic conventions used in the corpus were reasonably consistent, although they represent significantly more variation than what one finds in MSF texts. Some older materials might have been harder to handle, and some more recent materials might have been easier but less interesting to process.

An extract of the 1642 translation of the Bible (B1-Jes-3:11–3:12), along with its modern translation[3] (1992), is given in Table 1 with some notes on the structural differences between them.

Table 1: A small extract of Biblia 1642 and the same passage in modern translation

OLF	MSF	Note
Mutta woi jumalattomita:	Voi jumalatonta!	missing word
sillä he owat pahat	Hänen käy huonosti,	PL vs. SG, different verb and construction
ja heille maxetan	hänelle tehdään	different verb
nijncuin he ansaidzewat.	niin kuin hän itse teki.	different construction
Lapset owat minun Canssani waiwajat	Kansani valtiaat ovat lapsia,	different
ja waimot wallidzewat heitä.	ja naiset hallitsevat sitä.	different verb
Minun Canssani	Kansani,	
sinun lohduttajas häiridzewät sinun	sinun opastajasi vievät sinut harhaan,	different verb and case
ja turmelewat tien jota sinun käymän pidäis.	he ovat hämmentäneet askeltesi suunnan.	different verb construction

[1] http://kaino.kotus.fi/

[2] The parts VT-4 and UT-1 refer to the files available in the Kaino service.

[3] http://www.evl.fi/raamattu/1992/Jes.3.html

There are many kinds of differences between the translations. Some reflect orthographic conventions which have changed meanwhile, such as using `w` instead of `v` and sometimes a single letter `a` instead of a double `aa` for a long vowel. OLF of those days had more features from the western dialects than MSF. The language itself has also changed meanwhile and continues to change. The changes are both phonological and morphological: the OLF texts often omit word final letters. The use of ending allomorphs was then quite different and has changed significantly even during the last fifty years, as one can see by comparing *Nykysuomen sanakirja* (Sadeniemi 1951–1961) and *Kielitoimiston sanakirja* (Grönros & Kotimaisten kielten tutkimuskeskus 2006).

One can also find some words in the corpus that are not used in the MSF, and some familiar words are used in another sense. The present study did not try to solve such problems which concern the vocabulary and the lexicon. In this study, only phonological, morphophonological and to some extent the allomorphic differences are addressed.

2. Representative example words

It is obvious from the above examples that one cannot align the Biblia 1642 with the modern translation word by word directly, because the translations are so far apart from each other. Instead of statistical word alignment and large sets of words, we use a fairly small set of carefully chosen good quality examples.

We started from the list of all word forms which occurred at least six times in the corpus. The list was browsed and some 180 example words were picked. The words were chosen so that there were a few examples from each type of systematic differences between OLF and MSF written forms of the Finnish language. Figure 1 (overleaf) shows a fragment of the word forms.

It was important that the list of example words would cover all common systematic differences between the MSF and the OLF forms, including orthographic and morphophonological ones.

```
caupungihin
caupungijn
corwes
cuckoi
cuolemaan
cuoleman
cuolluitten
cuulitta
cuulcan
kärsimän
```

Figure 1: Some of the selected example words from the corpus

3. Example word pairs

The next step was to associate each of these OLF word forms with its likely MSF counterparts. The possible MSF forms corresponding to each OLF form were addded, see Figure 2. If the OLF word form could correspond to several MSF word forms, the OLF form was repeated, see `cuoleman` and `kärsimän` below. The relation between the OLF forms and the MSF forms is inherently many-to-many, i.e., one modern form may correspond to several different old forms, and an old form may correspond to several distinct modern forms. Rules must permit some variation but still constrain the possibilities to a minimum.

```
kaupunkiin:caupungihin
kaupunkiin:caupungijn
korvessa:corwes
kukko:cuckoi
kuolemaan:cuolemaan
kuoleman:cuoleman
kuolemaan:cuoleman
kuolleitten:cuolluitten
kuulitte:cuulitta
kuulkoon:cuulcan
kärsimän:kärsimän
kärsimään:kärsimän
```

Figure 2: The selected OLF example word forms with their corresponding forms in MSF. The MSF form is to the left of the colon and the OLF form to the right.

4. Character by character alignment

The above example word pairs are not usable for our purposes as such, because the OLF and the MSF word forms are sometimes of different length. The OLF form often omits a final vowel, reduces long vowels into short ones and shortens geminate consonants, but sometimes geminates a consonant or adds a vowel etc.

Therefore, we must add some zero symbols as necessary so that the similar letters correspond to each other, first letter in the MSF word to the first letter in the OLF word etc. If the MSF word is longer than the OLF word, one must add one or more zeros to the old word in order to make the letters correspond to each other. Correspondingly, if the modern word is shorter, the zeros have to be added in it. The goal is that the letters in all corresponding positions would become similar. Zeros are added as necessary, but sparingly, e.g., as in `kärsimään:kärsimäØn` 'to suffer'.

It must be stressed that a real character is used as a zero instead of an epsilon, an empty string or its representation `0` in the XFST regular expression language. For practical reasons, the Danish `Ø` was chosen as the zero symbol in rules and examples consistently in this article.[4]

The exact positions of the inserted zeros are as important as is the selection of the examples. The positions of the zeros determine what kinds of character correspondences we have. One must describe each correspondence with a rule, so the grammar may change a lot by changing the positions of the zeros a bit. In particular, any poorly positioned zero would force us to write more rules, and possibly very inadequate rules. The proper alignment also affects how well the grammar can apply to the rest of the corpus.

Letters representing similar (or identical) sounds ought to be matched with each other. Matching very different ones, e.g., consonants with vowels, must be avoided.

The initial insertion of the zeros was made manually using one's linguistic intuition as a guideline. Once the zeros were in place, we converted the pairs of words into sequences of pairs[5] of letter pairs as shown in Figure 3 where pairs with identical letters are printed as a single letter and pairs of corresponding non-identical letters are separated by a colon.

[4] Many finite-state tools interpret the digit 0 as a null string or epsilon. Often all traces of such a null string are lost in finite-state operations. In the two-level framework, this is not desired, and it is safer to delete the zero symbols `Ø` explicitly when desired.

[5] The conversion was done simply by:

```
hfst-strings2fst -i new-old-words.text |
hfst-fst2strings -X print-space -X print-pairs -o new-old-pairs.text
```

```
k:c a u p u n k:g i Ø:h i n
k:c a u p u n k:g i i:j n
k:c o r v:w e s:Ø s a:Ø
k:c u k:c k o Ø:i
k:c u o l e m a a n
k:c u o l e m a:Ø a n
k:c u o l e m a n
k:c u o l l e:u i t t e n
k:c u u l k:c o:Ø o:a n
k:c u u l i t t e:a
k ä r s i m ä n
k ä r s i m ä:Ø ä n
```

Figure 3: Some example words with zeros added and aligned letter by letter and shown as a sequence of letter pairs

Once we have the aligned pairs, we compute a list of different pairs and their frequencies as in Figure 4. The pairs end up as the declaration of the alphabet in the two-level rule grammar. The frequencies guide the authoring of rules and can be directly used for weighting alternative analyses.

```
158 a      2 e:ö    5 j:Ø 130 n     84 s     60 u      5 ö
 39 a:Ø   22 e:Ø   23 k    61 o      2 s:n    4 u:Ø    1 Ø:d
  1 b      2 f:p   53 k:c   2 o:a   16 s:z    1 v      1 Ø:e
  9 d      3 g     12 k:g   7 o:Ø   16 s:Ø    1 v:f    3 Ø:g
 97 e     26 h      6 k:x  30 p    102 t      2 v:g    4 Ø:h
  3 e:a  122 i     54 l     5 p:b   29 t:d   34 v:w    1 Ø:i
  3 e:i    4 i:j    1 l:Ø   1 p:w    1 t:l   17 y      1 Ø:n
  1 e:u   12 i:Ø   36 m     1 p:Ø    1 t:r    1 y:Ø    1 Ø:s
  1 e:ä    6 j      3 m:Ø  30 r      1 t:Ø   44 ä      4 Ø:t
                                             22 ä:Ø
```

Figure 4: Frequencies of the letter pairs found in the aligned example words

5. Automatic alignment

One may add further examples at the later stages of the research. One may also want to remove some examples, if they turn out not to represent any general patterns. To facilitate the maintenance of the collection of examples, an automatic character by character alignment was constructed, see also Koskenniemi (2017). Such an automatic procedure for character by character alignment is expected to be useful for other purposes, as well, including computational historical linguistics where it can be used in relating cognate words, cf. Koskenniemi (2013a).

The International Phonetic Alphabet (IPA) presents a general taxonomy for vowels and another for consonants, both based on the articulatory features of sounds. This taxonomy and the features can be utilised in computing approximate distances between sounds. Alphabetic scripts of MSF and OLF can be characterised quite well using the articulatory features of the IPA. For our purposes, only a subset of all features permitted by the IPA is needed.

A short Python script (see Appendix 4) was written for building a weighted finite-state transducer (WFST) out of the IPA features for the letters. For two-valued features, and for the tongue height of vowels and for the place of articulation of consonants, an ad hoc numeric value was assigned to each position.[6] The distances were computed by adding the absolute values of the differences in each feature. Insertions and deletions of letters were all given a constant fairly long distance. In addition to these systematically computed distances, some individual distances were set. These were needed e.g., in order to guarantee a unique treatment of the shortening of double vowels or consonants. Otherwise one could delete either of the two, and there would be no difference in the overall sum of distances. Thus, a few extra items were added in the distance calculation so that it is always the latter letter of the two that is deleted if any (e.g., `a a:Ø` rather than `a:Ø a`). Ambiguities caused by the orthographic conventions, e.g., between `k:x s:Ø` and `k:Ø s:x` and gemination (adding a second identical consonant after instead of before the existing one) were resolved in a similar manner.

A Python script was written, see Figure 5 (overleaf), to implement the actual alignment. The script uses the WFST for distances that was created as discussed above. The script reads an example word pair `(w1, w2)` at a time, converts the MSF word `w1` into a FST and inserts zero symbols Ø freely to it. The same is done for the OLF word `w2`. Then, `w1` is composed with the alignment WFST `align` and `w2`: `w1/Ø .o. align .o. w2/Ø`.

Out of the many possible string pairs that the resulting WFST represents, only the one with the smallest weight is taken and printed. When testing the alignment procedure, one can assess the relative success of each aligned pair of words. Each pair of words gets a score as the sum of all

[6] The actual process of aligning appears not to be sensitive to the choice of the distances among vowels and among consonants as long as consonants and vowels are not allowed to correspond to each other (with the exception of semivowels). It would worth while to find well motivated distance measures, maybe using data from historical linguistics.

```
import sys, io, fileinput
import libhfst
tok = libhfst.HfstTokenizer()
algfile = libhfst.HfstInputStream("chardist.fst")
align = algfile.read()
for line in sys.stdin:
   (f1,f2) = line.strip().split(sep=":")
   w1 = libhfst.fst(f1).insert_freely(("Ø","Ø")).minimize()
   w2 = libhfst.fst(f2).insert_freely(("Ø","Ø")).minimize()
   w1.compose(align).compose(w2)
   res = w1.n_best(1).minimize()
   paths = res.extract_paths(output='text')
   print(paths.strip())
```

Figure 5: Python script for aligning words letter by letter

character pair correspondence weights. Very high total weights indicate untypical pairs of characters which may sometimes be an error in the example word pair.

All finite-state functions that were needed for the script were available in the HFST-Python integration. This particular operation appears to be clumsy to perform using the standalone programs or XFST or Foma.[7]

6. Writing the two-level rules

For a more detailed description of two-level rules see Beesley & Karttunen (2003) and Karttunen et al. (1987). For the method of finding contexts for rules, see Koskenniemi (2013b). The rules to be written in this project have a common alphabet which consists of the letter pairs shown above in Figure 4. We have to write a two-level rule for each non-identical pair (unless there is just one alternative, or if we let all alternatives be allowed anywhere). The rules may be written in any order one finds convenient. Let us start with the pair `e:a`. Gnu Emacs was used for editing of test examples, rules and all other files. The Emacs command `Occurs` was thus available and used for extracting the right kind of information from the examples in letter pair format as in Figure 6.

These OLF word forms sound like some dialectal forms found even today. It was deduced that the correspondence `e:a` was restricted to two

[7] One can do it using the HFST command line programs by converting first the MSF words into a sequence of FSTs by hfst-strings2fst, the same for OLF words, and then composing the sequences element by element with the alignment FST.

```
3 matches for "e:a" in buffer: new-old-pairs.text
      71: k:c u u l i t t e:a
     141: t u l e t t e:a
     142: t u l i m m e:a
```

Figure 6: Occurrences of `e:a` in the examples

personal plural endings in verbs. Any other MSF word forms ending in `e` do not have OLF forms with `a` instead. Any letters `e` inside the MSF words are likewise unaffected by this alternation. This rule has no access to the grammatical features, it relies on patterns consisting of letters. Thus, the following rule in Figure 7 was written.

```
"e:a" e:a => [t t | m m] _ .#. ;
!                    k:c u u l i t t e:a
!                    t u l i m m e:a
```

Figure 7: Two-level rule which restricts the positions where MSF `e` may correspond to `a` in OLF

By convention, the examples based on which the rule was designed, were always included as comments to the rule. According to the conventions of the two-level rules, see Karttunen et al. (1987), this rule says that the pair `e:a` may occur only if preceded by `tt` or `mm` and is at the end of a word. Only the context restriction (`=>`) is used, not the double arrow[8], because there are some words where the stem ends similarly, e.g., `lumme` or `amme` where the final vowel does not change. Even the best and most obvious rules are bound to be ambiguous as long as one only has the surface representations available without any morphological or grammatical knowledge.

One can test the first rule right away after it has been written, as will be explained in the next section. Experienced two-level grammar writers often design a few rules before they test them. So, let us study another letter pair `s:Ø` before we proceed to testing, see Figure 8 (overleaf).

It is easy to see two patterns here. A double `ss` in MSF is reduced to a single `s` in OLF and `ks` in MSF words is represented as `xØ` in OLF. Thus, we need a rule with two context parts as in Figure 9 (overleaf).

Each rule is then compiled into a FST using the two-level compiler `hfst-twolc`. All rules together form the two-level grammar which

[8] A double arrow `<=>` rule would require that the change is obligatory in the given context.

```
16 matches for "s:Ø" in buffer: new-old-pairs.text
 14:  e d e s s:Ø ä
 25:  h a a:Ø h d e s s:Ø a:Ø
 26:  h a a k:x s:Ø i
 31:  h e n g e s s:Ø ä:Ø
 37:  h y v:w ä k:x s:Ø i
 52:  k:c a n s s:Ø a n s a:Ø
 60:  k:c o r v:w e s s:Ø a:Ø
 80:  m u r h e e:Ø l l i s e k:x s:Ø i
 83:  n i i:j s s:Ø ä
119:  s e a s s:Ø a:Ø
126:  s y d ä m e s s:Ø ä n s ä:Ø
127:  s y n a g o g a s s:Ø a:Ø
129:  t a p p a a:Ø k:x s:Ø e n s a:Ø
150:  u n e s s:Ø a:Ø
160:  v:w a p a a:Ø k:x s:Ø i
172:  y k:x s:Ø i n ä n s ä:Ø
```

Figure 8: Occurrences of s:Ø in the examples

```
"s:Ø" s:Ø => s _ ;
!                     e d e s s:Ø ä
                     s e a s s:Ø a:Ø
                  :x _ ;
!                     h a a k:x s:Ø i
```

Figure 9: Two-level rule for restricting the deletion of s

is compiled into a sequence of such rule transducers. If one has forgotten or mixed some punctuation in the rules when writing the grammar, there will be error messages with a pointer to the probable location and cause of the error. The grammar writer is expected to correct the error and recompile.

7. Validating the rules against examples

There is a facility for testing two-level grammars. There is a special program, `hfst-pair-test` which checks whether the grammar accepts all examples given as sequences of letter pairs. The same `Makefile` which compiles the rules, performs this check right away. The program reports any inconsistencies, e.g., character pairs occurring in other contexts than those allowed by the rules or misaligned words resulting in character pairs not allowed by the grammar.

Two familiar concepts from information retrieval are used here with a specific interpretation. *Recall* means here the proportion of OLF words that will get the correct MSF word among the results of the analysis (no matter how many wrong alternatives were produced). *Precision* means here the proportion of correct MSF results among all proposed results for a set of OLF words, e.g., all word tokens in the corpus. Recall and precision can equally well be used for the inverse relation, i.e., from the modern words to the old words.

One ought to remember that the testing of pair string examples only detects problems where the rules are too restrictive. Initially, before we have any rules, all examples would pass the check. Using just a few rules, one could retrieve all old forms for a modern form (as long as they participate in those alternations that were present in the examples). But the initial grammar has a very poor precision. A modern word corresponds to very many (possibly infinitely many) old words and vice versa. As we write more rules in our two-level grammar, the recall can only decrease, but every new rule improves the precision.

If one finds new types of regularities during the process, one ought to add new word pairs to the examples. New letter pairs can then be introduced, aligned and tested in the examples.

8. Standalone testing of the grammar

When one has rules for all letter pairs, the two-level grammar can be tested in a new manner. One can now generate tentative OLF forms from the MSF ones. One gets several results per each modern word. Using unweighted rules, all results of such generation are equal. There is one trivial weighting that can be used here to prioritise resulting words more likely: to use the statistics we have from the example words as in Figure 4. A short Python script is used for computing a WFST from the frequencies. Intersecting the weighted transducer with intersected rule FST gives us a new rule WFST. This one can be safely tested by inputting MSF words to it and selecting at most N, say 20, best results. If the correct one is among the top results, the rules seem to do the right thing. See the transcript in Figure 10 (overleaf) where one can see what the grammar generates out of a few modern words.

The weighted rule transducer can be inverted and thereafter tested in the same way. In the present project, the mapping from OLF to MSF

```
$ hfst-strings2fst | hfst-compose -2 intro.fst | \
  hfst-compose -2 new2old-one-w.fst |\
  hfst-compose -2 delete.fst | hfst-project -p output | \
  hfst-fst2strings -w -N 20

>>sija
sija         1.86035
sia          2.12402 +
>>sokeat
sokeat       3.84277
sokiat       8.85742 +
>>ruoskitte
ruoskitte    4.18848 +
ruoskitt     6.3291
ruoskitta    9.20312 +
ruoskite    10.8613
ruoskit     13.002
```

Figure 10: Testing how the plain rules generate tentative OLF word forms out of MSF word forms. The MSF word form as input is marked with `>>` and the correct results are marked with a plus sign (`+`).

words is expected to be more ambiguous than the other direction. Thus, the weighting is useful in checking the production of candidate modern forms. Figure 11 shows the 20 first results generated from an old word `isäm` 'our father' out of the total of 32 results.

For some other OLF words, there will be many more results, e.g., for `cullainen` 'golden', more than 300 results were produced. Even as such, the mapping might be useful in indexing or searching a corpus. One may easily produce a transducer `oldwords` which accepts exactly the word forms in the corpus. Composing the mapping `new2old` used in Figure 10 with `oldwords` could be quite useful. One could build a search facility on this basis which would use modern word forms as search keys and expand it according to `new2old` and do the actual search using the existing OLF words the mapping gives.

It would be impractical to use the above method in existing concordance programs, as it would require the inclusion of all alternatives, even the nonsense modern "word forms" in the index. However, nothing would prevent us from using `new2old` in a front end processor to traditional concordance programs such as Korp, described in e.g., Borin et al. (2012).

```
$ hfst-strings2fst | hfst-compose -2 intro.fst | \
 hfst-compose -2 old2new-one-w.fst | \
 hfst-compose -2 delete.fst | hfst-project -p output | \
 hfst-fst2strings -w -N 20

>>isäm
isäm       1.23633
isääm      2.94141
isäme      3.82129
issäm      4.31348
iisäm      4.84863
isääme     5.52637
issääm     6.01855
iisääm     6.55371
issäme     6.89844
isämme     7.40625 +
iisäme     7.43359
iissäm     7.92578
issääme    8.60352
isäämme    9.11133 +
iisääme    9.13867
iissääm    9.63086
issämme   10.4834
iissäme   10.5107
iisämme   11.0186
issäämme  12.1885
```

Figure 11: A test where we see the first 20 results that the inverted rules generate out of one OLF word *isäm*. The correct results are marked with a plus (+).

9. Combining the grammar with OMORFI

As we noticed above, the rules are quite ambiguous when generating tentative modern word forms from an OLF word form. We have a lot of candidates, among which the correct one is hidden. Most of the noise words are non-words in MSF. Thus, it is a natural idea to filter the noisy output of the rules using a spell-checker for MSF.

OMORFI is a finite-state morphological analyser which is open source and freely available, cf. Pirinen (2015). It uses the same HFST tools, so it was easy to combine it with other transducers used in this study. For further information on the HFST morphological tools, see e.g., Lindén et al. (2011). OMORFI is distributed both as source code and as binary

FSTs.[9] The source form consists of more than 300 files and appears fairly complicated. More than a dozen Makefiles are needed for building the FST that recognises Finnish word forms. Therefore, it was easier to use the binary transducer which comes with the package even if there would have been an obvious need to modify the lexicon and rules to better suit the needs of this project.

The transducer `finnish-analyze.fst` takes a Finnish word form as its input and outputs its analyses as a combination of a base form and the morphosyntactic features characterising the grammatical form, e.g., as in Figure 12.

```
$ hfst-strings2fst |\
    hfst-compose -2 finnish-analysis.fst |\
    hfst-fst2strings
        >>kuutamoilta
        kuutamoilta:kuutamo N Abl Pl
        kuutamoilta:kuutamo#ilta N Nom Sg
```

Figure 12: Morphological analysis using plain OMORFI. Two outputs are generated from the input *kuutamoilta* which is either 'from moonlights' or 'moonlight' + 'evening'. Note the word boundary in the second result.

The morphosyntactic features are not needed for the filtering of the noise words from the set of candidates that the rule transducer generates. Only the input side of the transducer is needed for the selection of acceptable word forms of MSF. One can simply drop the output part and keep the input side of the analysis FST.[10]

The mapping all the way from OLF word forms into valid MSF word forms is the composition of four transducers in a sequence, see Figure 13. One may run these as a run-time pipeline using separate HFST programs or one may compose them in advance for efficiency.

The combination of the steps in Figure 13 does roughly what was expected. If we feed the OLF words in Figure 1 to it, each old word will be expanded to several possible MSF word candidates, and the analyser will then filter out all but those candidates that it considers acceptable MSF word forms, as is seen in Figure 14.

[9] The FSTs distributed were in a so called fast lookup form. One can convert them back to the HFST standard form and then modify and manipulate them for the needs of this project.

[10] Projection is made using the command `hfst-project -p input`.

OLF word form

↓

`intro.fst`

↓

OLF word form with zeros added in all possible ways

↓

`old2new.fst`

↓

candidates for MSF word forms with zeros

↓

`delete.fst`

↓

candidates cleaned of all zeros

↓

`finnish-analyze-surf.fst`

↓

valid MSF word form candidates

Figure 13: Producing MSF word form candidates out of OLF word forms

```
caupungihin [kaupunkihiin, kaupunkiin]
caupungijn [kaupunkiin, kaupunkiini]
corwes [korvessa]
cuckoi [kukko]
cuolemaan [kuolemaan, kuolemaani]
cuoleman [kuolemaan, kuolemaani, kuoleman, kuolemana,
          kuolemani, kuolleemman]
cuolluitten [kuolleitten, kuolleitteni]
cuulcan [kuulkoon]
cuulitta [kuulitta, kuulitte]
kärsimän [kärsimän, kärsimäni, kärsimään, kärsimääni]
```

Figure 14: Analyses of some example words using the two-level grammar and filtering with OMORFI

It is to be noted that most of the modern word forms offered by the sequence are quite acceptable. In particular, all correct interpretations that we wanted are present. In addition to the desired results, there are some artificial words. One of them is the very first result `kaupunkihiin` 'to the city' which looks odd. It turns out to be a compound of `kaupunki` 'town' and `hiki` 'sweat' which is a nonsense word. Another extra result is `kuulitta` 'you heard', is also an odd compound of `kuu` 'moon' and `litta` (a children's play e.g., with a ball).

The number of compound boundaries in a word form would be useful as a criterion for excluding less likely analyses. Unfortunately, when using OMORFI, this information is only available when one reduces the word forms all the way to their base forms. With some Python scripting and processing, the knowledge about the number of compound boundaries can be used at the right place. One first produces a list of all pairs where the first component is the OLF word form and the second component is the analyses OMORFI accepts from the many candidates that the rules propose. The following pairs are in the long list:

```
aitais:aitaiisi
aitais:aitaisi
aitais:aitasi
aitais:aittaiisi
aitais:aittaisi
aitais:aittasi
```

The next step is to analyse the right parts again which were once already accepted by OMORFI, and we get a list containing entries like the following:

```
aitaiisi:aita#iisi A Pos Nom Sg
aitaisi:aidata V Cond Act ConNeg
aitaisi:aidata V Cond Act Sg3
aitaisi:aita#isi N Nom Sg
aitasi:aidata V Pst Act Sg3
aitasi:aita N Gen Sg PxSg2
aitasi:aita N Nom Pl PxSg2
aitasi:aita N Nom Sg PxSg2
aittaiisi:aitta#iisi A Pos Nom Sg
aittaisi:aitta#isi N Nom Sg
aittaisi:aitta N Gen Pl PxSg2
```

From these pairs, we only use the number of word boundaries # in the stem that is on the right. For each surface form we store the least number of boundaries its base form analyses have. The first one has only a compound analysis, so it gets the count 1. The next, `aitaisi` 'of your barn(s)' or 'of your fence(s)' has three analyses, two without boundaries and one with one boundary, so it gets the count 0:

```
aitaiisi 1
aitaisi 0
aitasi 0
aittaiisi 1
aittaisi 0
aittasi 0
```

Now one can return to the processing of the result pairs where the left part is the OLF word and the right part is a word form proposed by the rules and accepted by OMORFI. For each OLF word, we now have a list of candidate MSF words. We can fairly safely drop some candidate MSF word forms by using their compound boundary count as computed above. We throw away all candidates which have more compound boundaries than the one that has the least number of them. Thus, we start from the following list of modern forms for the OLF word form `aitais`:

```
aitais 1 [aitaiisi, aitaisi, aitasi, aittaiisi, aittaisi, aittasi]
```

According to the counts we computed, the first and the fourth have a boundary count 1 and the rest has no boundaries. Thus, we drop the first and the fourth, and get the final result which now contains only acceptable words and no artificial constructions:

```
aitais 1 [aitaisi, aitasi, aittaisi, aittasi]
```

This processing sounds complicated,[11] but it is motivated by the fact that OMORFI produces a lot of extra analyses using its liberal compounding mechanism. Anyway, the Python script which does the trick, is short, fast and straightforward.

10. Reducing to the base forms

Normally, OMORFI reduces word forms to their base forms, and base forms would be often even better for searching and indexing than the word forms themselves. Thus, in parallel to the operations in the previous section, the candidate MSF word forms were filtered and reduced to their base forms. This list had the same kinds of problems with the liberal compounding of OMORFI as we saw in the previous section. The artificial

[11] It would also be possible to handle the stem counts using WFSTs. One may convert the list of MSF forms and compound part counts into a weighted transducer which accepts the modern forms and use the weight as a criterion for exclusion.

compounds could be removed in the same way, in fact more easily as the compound boundaries were present in the base forms directly. Before the filtering, the results for a base form `alendamisest` 'from lowering' looked like the following:

```
alen#da#miss#eesti   'sale'+'da'+'miss'+'Estonian'
alen#da#miss#este'sale'+'da'+'miss'+'obstacle'
alentaa  'to lower'
alentaminen  'lowering'
alentamis#eesti  'lowering'+'Estonian'
alentamis#este   'lowering'+'obstacle'
alen#tamminen'sale'+'made of oak'
alen#tammis#eesti'sale'+'made of oak'+'Estonian'
alen#tammis#este 'sale'+'made of oak'+'obstacle'
```

Again, the filtering program considers this set of candidate MSF base forms. It finds two candidates with no compound boundaries and seven with one or two boundaries. The program throws away those seven and keeps the two. So the result for `alendamisest` becomes:

```
alentaa
alentaminen
```

11. Tuning the two-level rules

At this point the rules have been tested against the examples and they have been used separately for some manually typed words in order to assess the precision of the rules, i.e., how many unwanted analyses they produce. We have tools for reducing OLF word forms into MSF word forms and also to MSF base forms. Now one can see what the rules and OMORFI together actually do to the masses of words of the Biblia 1642 corpus.

One can expect that some rules are too permissive. This will show up as too many candidate MSF words. On the other hand, some rules might have too narrow context conditions, which will be seen as some OLF words left without the desired candidate words. It is also possible that some regular phenomena were not present in the example words. Then we have no applicable rules and many OLF words remain without the desired candidate MSF words. In the two first cases, we must consider revising the two-level rules we have written. In the last case, we must select further example word pairs and write yet another rule and test it.

In order to check what actually exists in the the Biblia 1642 corpus, three files were used: the source text itself, an alphabetical list of distinct OLF word forms in the corpus, and a list of reversed OLF word forms (sorted starting from the last character). Using the Gnu/Linux `less` and `egrep` commands, one got quick answers to questions such as: "Are there many other words similar to this one?" or "Is this really a form of the word I think?".

The tuning consumed more time than the writing of the initial two-level grammar. It was also more demanding because one must check that changes in rules do not have negative effects, such as dropping some desired candidate words which were previously correctly generated. For this purpose, the changes in the rules were always checked by producing a separate new list and comparing it against a previous full list of analyses[12]. If the differences were all for the better, then the new rules were accepted, and the new lists taken as the new benchmark for the following changes. Some of the new or lost analyses required checking from the corpus or the lists of old words as mentioned above.

The tuning required partial knowledge of the language in the corpus and was made by Kimmo Koskenniemi. An overall sense of present day Finnish and some familiarity with Finnish dialects seemed to be sufficient for finding generalisations and adequate context characterisations. Just one OLF word form (`käätyxi`, 'that has been turned' could not be interpreted by looking at the Biblia 1642 occurrences. One had to look it up in a more recent Bible translation.

All changes of the rules were automatically checked against the collection of hand-selected example word pairs. Any discrepancies were immediately detected and the rule violating some word pair was identified. After correcting the rule that failed, the rules were recompiled, retested and the full lists were recomputed. A handful of new example words were included in this process. The original and the new examples were used to test the rules thereafter at every cycle.

There appears to be no clear limit how long one can tune a grammar. After a certain level is reached the return of each cycle diminishes. Many of the remaining shortcomings could be better solved if one could have a different morphological analyser for Finnish. In particular, one would like to modify the compounding mechanism, make the derivational capacity more productive, and use a morphophonemic representation for MSF as the basis for rules. Then one would have access to many relevant conditions for determining the forms of the OLF.[13] Such a re-implementation of Finnish

[12] The checking was done using the Gnu/Linux `comm` program.

[13] OMORFI is open source and readily available for modifications. It was designed, however, without morphophonemes or explicit indication for alternating phonemes. OMORFI is very good for e.g., spell checking and even for generating inflected forms

morphological analysis would be motivated also when applying two-level methods to historical linguistics of Finno-Ugric languages, see Koskenniemi (2013a).

A couple of cases occurred where a new letter pair and an entirely new rule had to be established. That posed no major problem as long as the main principles of letter alignment and correspondences remained unchanged. With a few new example word pairs, there were no particular problems.

A common question that arose was to decide whether the rejection of MSF word forms was an error or a feature of OMORFI. The analyser is committed to obey the guidelines for word inflection as described in *Kielitoimiston sanakirja*[14] (2006) which is also available as a net service.[15] In most cases, a fifty years earlier norm of MSF would have better suited the needs of this project.

12. Evaluation of the mapping

The rules were developed using a set of example words. So the discussion of the success and the shortcomings of the mappings cannot be estimated by testing with the same words. One can assess how the mapping covers the vocabulary of the corpus by taking a sample of the list of all distinct word forms in the corpus, i.e., some 26,500 words. This list consist mostly of infrequent words. Half of them are hapax legomena, i.e., occurring only once. Less than 5,000 words occur more than five times in the corpus. Two 100 word samples were selected, one out of the full list of distinct word forms and another from a list consisting of word forms occurring at least six times in the corpus. Both samples were made out of the respective total list by first skipping some entries and then proceeding with even intervals (the length of the list divided by 100). A third sample was made from the running text.

12.1. Proper nouns and abbreviations

The Biblia 1642 corpus contains plenty of proper names, biblical and other. Names of persons and places occur typically fairly few times and only

e.g., in machine translation. Modifying it for the present purposes would be as difficult as building a new analyser.

[14] The dictionary of the Institute for the Languages of Finland.

[15] http://www.kielitoimistonsanakirja.fi/

within a short passage of text. There are two kinds of problems concerning them. Dictionaries lack most of them, so the filtering could not work properly. Most proper names are unlike normal Finnish words, and the orthography used in writing them differed from that of normal OLF words. Proper names are often written as in Swedish or German and not adjusted to Finnish.

By mistake, some material, such as references to other parts of the Bible remained in the corpus although the intention was to exclude them all. This happened probably because such markings had more variable forms than was expected. The abbreviations included in this way are not valid OLF words and not a target of this study.

Thus, the proper nouns and abbreviations do occur in the samples, but could be ignored in the results. Proper nouns, and many other words were written with capital letters in Biblia 1642. In addition, capital letters are found in the corpus in unusual positions, e.g., both as the first and the second letters. Precise normalisation of the corpus was not a goal of this project, so nothing was done beyond forcing all text to lower case.

12.2. Words occurring more than five times

The result of testing a sample of 100 word forms from the list of word forms occurring at least six times in the corpus is given in Appendix 1. The following is a summary of the results with this sample:

- Eight biblical proper names or abbreviations were missed.[16]
- In addition, six OLF word forms were left without a proper analysis: an obsolete form `käätyxi` of the verb `käännetyksi` 'turned', 'converted'; slightly archaic inflections `löytty` 'found', `saatit` 'you might', 'you escorted', `sijpein` 'of wings', `vartioidzit` 'they/you guarded'; a word form `mixette` 'why not (you)' accidentally missing from OMORFI. One may assume that these and similar words would be correctly analysed, if the filtering morphological analyser could be modified so that it accepts older inflectional forms.

[16] Four of the eight would be accepted by OMORFI if given with a capital letter: proper names `Amoksen`, `Saul`, `Jerusalemista`, and the roman numeral `XI`. The proper name `Gedalia` would not have been in OMORFI, neither the two abbreviations `Cap` and `Ioh` accidentally included in the list.

12.3. All word forms in the corpus

The full list of the other 100 word sample which was taken from the total list of all word forms occurring in the corpus is given in Appendix 2. The following is a summary of the results:

- There were 11 proper names or abbreviations which were rejected: `Ahabin`, `Bath`, `Giledassa`, `Ismaelille`, `Kanaaneri`, `Kyrenestä`, `Magnus`, `Moph`, `Pet` (= Petrus), `Pilatus`, `Publiuksella`. See the discussion above.
- In addition, 16 words did not get the proper analysis, 10 of them occurred just once in the corpus. Two fairly frequent words (`ettäs`, 'you not', `synteins`, 'of their sins' were not accepted by OMORFI and therefore were lost, and so was `poismenit` 'you went away' which is nowadays written as two words.
- Out of the remaining 13 unanalysed words, one `onnettomudexen` 'to his misfortune' was analysed to the correct base form but not to the less frequent form actually used in the corpus.
- For three words, one could search further for a possible revision of the rules: `cauhiuttain`, `julgista` 'priech', not 'public', `tervehdimmä` 'we welcome(ed)'.
- Many missed OLF words might be better handled by revising the lexicon of the morphological analyser rather than the rules developed for this paper. There is no guarantee that rule revisions would really improve the recall. The OLF word `onnettomudexen =onnettomuudeksenne` 'to your misfortune' is a case in the point. It would be tempting to include a rule which would allow the recognition of the plural second person possessive suffix by deleting the two final letters. One ought to be careful, though: the rule would overgenerate because there are many other words ending similarly and not so many occurrences of this suffix.[17] One would need access to the morphophonological level of MSF in order to describe this suffix accurately.
- Altogether some 165 analyses were produced for the 100 word forms. About 79 of them were exact matches to the actual word in the

[17] It is possible to modify the endings in OMORFI although one has to make changes in many places of the lexicon.

corpus. Ten proposed analyses were unacceptable (six as artificial compounds, four as guesses for proper names).

12.4. Sample of word tokens from the running text

The two above tests estimated how well the method covers the vocabulary. Another aspect is, how well the method covers the text, i.e., how large a portion of word tokens in the running text would get a proper base form by which the place could be retrieved. For this purpose, a sample of 100 words was made, starting with a small offset and stepping through the text at equal intervals. A summary of results with this test:

- Four word tokens were proper names: `Babelin`, `Efraimin` , `Israelin`, `Maria`; two were abbreviations: `Reg`, `XXI` and were left without a correct analysis.
- Three tokens were left without a proper analysis: `iohtunut` 'caused by', `murehitit` 'you worried', `sijhenasti` 'till then'. The `j:i` is a very rare correspondence, and the last two ones are ungrammatical in MSF.
- The remaining 91 words of this sample were given, among others, the correct analysis.

One may speculate that frequent words are more common in samples of running text than in samples from lists of distinct word forms. Therefore, it would be expected that the rules and OMORFI perform better with such samples.

13. Conclusion

On the whole, the authors consider the precision and recall of the combination of the two-level rules and OMORFI successful, somewhat better than was expected. Spending more time with the rules and tuning the context conditions would not have a significant effect on the performance. By making the conditions looser, one may improve the recall at the expense of precision. With some manually compiled lists and by paying attention to the capital letters, one could handle the proper names much better.

The most promising line of development would be to build a different type of morphological analyser. The OLF form like `jalgat` 'feet' corresponds to the MSF surface form `jal∅at`. Adding a potential `g` in all possible places seems a bad idea. The morphophonemic representation of the modern form could be something like `j a l k∅ a + t`. Relating `g` to the morphophoneme `k∅` describes the phenomenon more logically.

Acknowledgements

We are grateful to the Institute for the Languages of Finland for the free availability and use of the Old Literary Finnish materials which was necessary for this project. We are also grateful to the FIN-CLARIN project which has implemented the HFST software and offers technical support for it.

The division of labour between the authors was that Pirkko Kuutti selected the material for the corpus and prepared the set of examples and checked the judgements in the final results. Kimmo Koskenniemi did all other tasks.

Appendix 1: Sample of words occurring at least 6 times

This sample was made from the word forms occurring at least six times in the corpus. The list was divided in 100 parts of equal length and then the 42nd word from each part was selected. This sample was not used for writing or tuning the rules. The rules were not changed after this sample was processed and this article was written.

The analysis of the sample was manually checked looking up the passages in the corpus for judging what the original word form stood for. Decisions were inserted in the list using the following markings: names or abbreviations are marked with a preceding (§). They were left outside the present study because OMORFI would not cover them in any case. OLF words which were left without any correct analyses are marked with a preceding (@) sign. After an equal sign (=), a desired result is given, i.e., a result one would wish that the analysis would produce. Those MSF word forms which were attested to be correct are marked with a plus sign (+). The results which were considered to be wrong are marked with an asterisk (*). The remaining unmarked MSF word forms in the results are formally possible but not attested in the Biblia corpus.

```
ajasta 22 +ajasta, ajastaa
§ amoxen 6 =Amoksen
apostoleille 8 +apostoleille
asti 178 +asti
autuuden 21 +autuuden, autuuteen, autuuteeni, +autuuteni
cadzos 20 +katsos
callis 7 kalliisi, kalliissa, +kallis
§ cap 150 =abbreviation (not part of the text)
catumattomudens 9 kaatumattomuuteensa, kaatumattomuutensa,
                  katumattomuuteensa, +katumattomuutensa
§ cesareaan 11 =Kesareaan
```

```
colmas 26 +kolmas
costaman 10 koostamaan, koostamaani, koostaman, koostamani,
   +kostamaan, kostamaani, +kostaman, kostamani
cuitengin 328 +kuitenkin, kuittenkin
cunnias 9 kunniaasi, +kunniasi, +kunniassa
cuulemma 7 kuulemma, +kuulemme, kuulleemme, kuullemme
duomidzeman 20 +tuomitsemaan, tuomitsemaani, +tuomitseman,
               tuomitsemani
egyptiläisten 10 +egyptiläisten, egyptiläisteni
engelille 8 +enkelille
epäjumalain 22 +epäjumalain, epäjumalaini
että 2808 +että
§ gedalia 7 =Gedalia
harwat 15 +harvat
heitän 9 +heitän, =heidät, heittäne, *heittäni, *heittään,
   *heitäni, *heitään
hetke 14 +hetkeä
huones 35 +huoneesi, +huoneessa
hywä 160 +hyvä, +hyvää, =hyvät
häwitetän 14 hävitettäne, hävitettäni, hävitettään,
   +hävitetään, hävitteettäni, hävitteettään
ihmisest 12 +ihmisestä
§ ioh 21 =abbreviation (not part of text)
itkemän 17 +itkemän, itkemäni, +itkemään, itkemääni
§ jerusalemist 58 =Jerusalemista
jolle 14 +jolle
judalaisista 12 +juutalaisista
jumalinen 9 +jumalinen, jumalineen, jumalineni
kedolla 53 +kedolla, ketolla
kircasta 7 kirkasta, +kirkastaa
kylään 7 +kylään, kylääni, kyyllään, kyylään, kyylääni
käsiwartens 14 käsivarteensa, +käsivartensa, käsivarttensa
@ käätyxi 21 =käätyksi (obsolete inflection pro 'käännetyksi)
laskeman 7 +laskeman, +laskemaan, laskemaani, laskemani
lewitat 11 +leviitat
luetan 14 luetan, +luetaan, luettane
lyödyxi 6 +lyödyksi
@ löytty 13 =löytty (obsolete inflection pro 'löydetty')
mailma 57 +maailma, +maailmaa
menewät 44 +menevät
miehens 11 mieheensä, +miehensä
@ mixette 6 =miksette (OK in MSF but OMORFI rejects)
muu 30 +muu
neljäkymmendä 16 +neljäkymmentä, neljääkymmentä
nimittä 12 nimittä, +nimittää
nähdä 88 +nähdä
oikein 87 +oikein, oikeine, oikeini
oma 34 +oma, +omaa
opetuslastens 40 +opetuslastensa
oxa 11 +oksa, +oksaa
```

```
paimen 19 +paimen
palwelusta 22 palvellusta, +palvelusta
parembi 11 +parempi
perkelestä 6 +perkeleestä
pidetän 14 pidettäne, pidettäni, pidettään, +pidetään,
           piteettäni, piteettään
pohjaisest 8 +pohjaisesta, pohjaisesti
prophetalle 26 +profeetalle
puolelle 19 +puolelle
päiwiä 9 +päiviä
päät 7 +päät, pääte
rangaisewa 18 +rankaiseva, rankaisevaa
ristinnaulidzit 6 ristiinnaulitsit, ristiinnaulitsitte,
                  +ristiinnaulitsivat
ruumis 10 +ruumis, +ruumiisi, ruumiissa, ruumissa
@ saatit 14 =saatit (old inflection pro 'saattoivat'), *saatit
sanani 45 +sanani, +sanaani
§ saul 8 =Saul
seuracunda 38 +seurakunta, +seurakuntaa
@ sijpein 10 =siipein (old inflection pro 'siipien')
sisäldä 11 +sisältä, sisältää
sucucunda 25 +sukukunta, +sukukuntaa, *suukukunta, *suukukuntaa
suus 9 +suusi, +suussa
synnyttä 13 synnyttä, +synnyttää
tahto 243 tahto, +tahtoa, +tahtoo
tapahtunut 65 +tapahtunut
tehkät 47 +tehkää
tie 15 +tie
toiseens 6 +toiseensa
tulella 35 +tulella, tulleella, tuulella, tuulleella
turmelit 6 +turmelit, turmelitte, +turmelivat
tyttäres 10 tyttäreesi, +tyttäresi, tyttäressä
täytti 11 +täytti
uscollinen 10 +uskollinen
waelsi 26 +vaelsi
waldacundain 15 +valtakuntain, valtakuntaini
wanhast 8 +vanhasta, vanhasti
@ wartioidzit 8 =vartioitsivat (old inflection pro 'vartioivat')
wertauxen 34 vertaukseen, vertaukseeni, +vertauksen, vertaukseni
wihollisen 15 viholliseen, viholliseeni, +vihollisen, +viholliseni
woi 223 +voi
wuorten 22 +vuorten, vuorteni
§ xi 13 =XI (not part of the text)
yljän 13 +yljän
ystäwä 6 +ystävä, ystävää
änellä 27 +äänellä
```

Appendix 2: Sample of all OLF word forms

This sample was taken from the full list of all distinct OLF word forms of the corpus, i.e., each word form appeared only once in the list no matter how many times it occurred in the corpus. The sample starts with the 86th word and proceeds with steps of equal length in the alphabetical list. The markings for proper names or abbreviations (§), words with no analysis (@), manually added interpretations (=), correct analyses (+) and completely irrelevant candidates for MSF words (*) follow the same principles as in the sample in Appendix 1.

```
§ ahabin 3 =Ahabin
@ alaidzen 1 =alitse
andimexi 1 +antimeksi
arpoja 1 +arpoja
asuwat 59 +asuvat
§ bath 3 =Bath
cahleis 5 +kahleissa
@ cananeri 1 =kanaaneri=Kaanaan asukas, *kanan-erie
carsi 2 kaarsi, +karsi, karsii
@ cauhiuttan 1 =kauheuttaan
@ cherubim 5 =kerubim=kerubi
colminaisuden 2 +kolminaisuuden, kolminaisuuteen,
    kolminaisuuteeni, kolminaisuutena, kolminaisuuteni
cotcatkin 1 kotkaatkin, +kotkatkin
cullastans 1 +kullastansa, kultastansa
cuolettaman 1 +kuolettamaan, kuolettamaani, kuolettaman,
    kuolettamana, kuolettamani
cuurnidzet 2 +kuurnitset, kuurnitsette
edestäm 8 +edestämme
eläwäin 1 +eläväin, eläväini
epäjumalista 2 +epäjumalista
@ ettäs 100 =ettäs (OMORFI)
§ gileadis 5 =Gileadissa
halkeisit 1 halkeisit, +halkeisivat, halkeisitte
hedelmälisest 2 +hedelmällisesti, hedelmällisestä
@ heräjä 1 =heräjä=herää, herääjä, herääjää
hopiaxi 1 +hopeaksi
hurscana 1 +hurskaana
häpiäs 13 +häpeäsi, +häpeässä, +häpeääsi
ihmisildä 17 +ihmisiltä
§ ismaelille 1 =Ismaelille
jalca 3 +jalka, +jalkaa
johdatan 3 +johdatan, johdattane
@ julgista 1 =julkistaa
jutteli 7 +jutteli
kelwatcon 1 +kelvatkoon
kijtoswirren 4 +kiitos-virren
kitans 1 kitaansa, +kitansa
§ kyrenist 1 =Kyrenestä
kätensä 1 +kätensä, +käteensä, kättensä
```

```
lainaxi 1 +lainaksi
laulun 3 +laulun, lauluna, lauluni, lauluun, lauluuni
lewollisest 1 +levollisesta, levollisesti
lohdutuxellans 1 +lohdutuksellansa
luotat 5 +luotat, luotaat, luotaatte, luotatte
lähikyläins 1 +lähikyläinsä
§ magnus 3 =Magnus=Suuri, *maa-gnuusi
@ medzäficunapuulle 1 =metsä-viikuna-puulle
miehest 4 +miehestä
§ moph 1 =Moph
muucalaisilda 1 +muukalaisilta
@ nautitcat 1 =nautitkaa=nauttikaa
nimiä 1 +nimiä, nimeä, nimeää
nurisewat 1 +nurisevat, nurissevat
@ ohrapion 1 =ohrapivon=ohrakourallisen, *ohrapioni
@ onnettomudexen 1 =onnettomuudeksenne, onnettomuudekseen,
    onnettomuudekseni
ota 102 +ota, oitta, otaa, +ottaa
pahenetta 1 +pahenette
paljastawat 1 +paljastavat
paransin 2 +paransin
peljännet 2 peljännet, +peljänneet, peljännette
§ pet 6 =Pet=Petrus=abbreviation, *peet
§ pilatus 58 =Pilatus, *pilattusi, *pilattuusi, *pilatussa
@ poismenit 1 =poismenit=pois menit
§ publiuxella 1 =Publiuksella
purpuraan 1 purpuraan, +purppuraan, purppuraani, purpuraani
päälimmäistä 1 +päällimmäistä
racastawanans 1 +rakastavanansa
rascautta 2 +raskautta, +raskauttaa
riemuhuudon 1 +riemuhuudon, riemuhuutona, riemuhuutoni,
    riemuhuutoon, riemuhuutooni
rucouxens 3 rukoukseensa, +rukouksensa
saamme 8 +saamme
saitte 4 +saitte
@ saphir 3 =safiiri
selitetyt 1 +selitetyt
sielä 1 +siellä
@ sisälmäisin 1 =sisälmäisiin=sisimmäisiin?
sotawäke 4 +sotaväkeä
suremmaxi 1 +suuremmaksi
@ syndeins 30 =synteinsä/syntiensä (OMORFI)
syöxemän 1 +syöksemän, syöksemäni, +syöksemään,
    syöksemääni
taitons 1 +taitonsa, taitoonsa, taittonsa, taittoonsa
taudist 3 +taudista, tautiista, tautiistä, tautista
@ terwehdimmä 2 =tervehdimme
todistaja 4 +todistaja, todistajaa
tottunet 3 +tottunet, tottuneet, tottunette
tunnustin 2 +tunnustin
```

```
tyhmäin 1 +tyhmäin, tyhmäini
töilläs 1 +töilläsi, töiltäsi
uscalda 4 +uskaltaa
wacudes 1 vakuudessa, vakuuteesi, +vakuutesi
waiwan 12 +vaivaan, vaivaani, +vaivan, vaivana,
    +vaivani
wallidzewat 7 +vallitsevat
warcaudella 1 +varkaudella
weidzet 1 +veitset
wialliset 2 +vialliset
wihollisillans 1 +vihollisillansa
wircaan 6 +virkaan, virkaani
wuoria 7 +vuoria
wääristä 4 +vääristä, vääristää
@ ylöllist 1 =ylöllistä=ilkeätä?
yxi 334 yksi
```

Appendix 3: Two-level rules

```
Alphabet
 a a:Ø b d e e:a e:i e:u e:ä e:ö e:Ø f:p g h i i:j i:Ø
 j j:i j:Ø k k:c k:g k:x l l:Ø m m:Ø n o o:a o:Ø p p:b p:w
 r s s:n s:z s:Ø t t:d t:l t:n t:r t:Ø u u:Ø v v:f v:g v:w v:Ø
 y y:Ø ä ä:Ø ö ö:ä ö:Ø Ø:d Ø:e Ø:g Ø:h Ø:i Ø:n Ø:s Ø:t ;

Sets
 Vowel = a e i o u y ä ö ;
 Cons = b c d f g h j k l m n p r s t v w x z ;

Definitions
 Suf1 = [n i: | n s a: | s i: | m m: e:] ;
 Suf2 = (k i n | k a :Ø n) ;
 aSuff =  ((a:) (n | Suf1) | i n | i Suf1 | l [l e|t a] (Suf1) |
           n | n a (Suf1) | s [s:|t] a: (Suf1) | t |
           [t:|:t] a (Suf1) | k:x s: e Suf1 | k:x s:Ø i) Suf2 .#. ;
 oSuff =  [k:x s: i | l l [a|e (e: n)] (Suf1) | n [a|e] (Suf1) |
           s [s:|t] a: (Suf1) | t | [t:|:t] a (Suf1) |
           t t e n | t t e Suf1] Suf2 .#. ;
Rules

"a:Ø" a:Ø => a: _ ;
!                    p a l a j a a:Ø
!                    r a a:Ø m a t t u
             :Cons e _ .#. ;
!                    k:c a i k:c k e a:Ø
             :Cons o a:Ø _ .#. ;
!                    h o l h o a:Ø a:Ø
             :Cons o _ [(a:Ø) .#. | :i :s | j [a|i] | :m |
```

```
                       :n .#. | t (t e:) .#. | :w | :∅* :x] ;
!                    k i r o a:∅ i s i t
!                    p u t o a:∅ v:w a t
!                    p u t o a:∅ m i s i l l a
!                    v a i n o a:∅ a:∅
!                    v:w a i n o a:∅ j a n i
!                    v:w a i n o a:∅ t t e:a
!             [n | s (s:∅) | s t] _ .#. ;
             [s (s:∅) | s t] _ .#. ;
!                    s e a s s:∅ a:∅
!                    a i k:c a n a:∅
!                    e v a n k:g e l i u m i s t a:∅
             i v:∅ _ t .#. ;
!                    a n t:n o i v:∅ a:∅ t
!                    s o t:d i:e i v:∅ a:∅ t

"e:∅" e:∅ => e _ ;
!                    i h m e e:∅ t
             _ .#. ;
!                    i s ä m m:∅ e:∅
             _ i t [t e n | a | ä] .#. ;
!                    a p o s t o l e:∅ i t t e n
             i s _ n [a | ä] .#. ;
!                    t o i s e:∅ n a
             a t _ r i [a | o] ;
!                    a t e:∅ r i o i t:d s:z i

"e:a" e:a => [t t | m m] _ .#. ;
!                    k:c u u l i t t e:a
!                    t u l i m m e:a

"e:u" e:u => l _ i t ;
!                    k:c u o l l e:u i t t e n

"e:ä" e:ä => n _ m ;
!                    e n e:ä m p:b ä ∅:t ä

"e:i" e:i => t a _ n .#. ;
!                    o p e t t a e:i n
             _ [a | ä] ;
!                    r u s k e:i a t

"e:ö" e:ö => .#. y l _ n ;
!                    y l e:ö n k:c a t:d s:z o

"f:p" f:p => _ ∅:h ;
!                    p r o f:p ∅:h e e:∅ t t:∅ a i n

"i:j" i:j => i _ ;
!                    n i i:j s s:∅ ä
```

```
"i:∅" i:∅ => i _ ;
!                    r u u m i i:∅ n
            o _ t ;
!                    o s o i:∅ t t i
            [n | s | s t] _ .#. ;
!                    n i m e s i:∅
!                    k:c o l m a s t i:∅

"i:∅ .#." i:∅ <= i _ .#. ;

"j:i" j:i => .#. o r _ a ;
!                    o r j:i a t

"ij:i∅" j:∅ => [k: | l | s: | t:] i _ [a aSuff | o i oSuff]  ;
!                    k:c a m a r i p a l v:w e l i j:∅ a ∅:t a
!                    k:c a u p i t:d s:z i j:∅ a t
!                    k:c u l k i j:∅ o i t a
!                    k:c u r k i s t e l i j:∅ a t
!                    h a k i j:∅ a t
!                    h a l t:d i j:∅ a
!                    h a l t:d i j:∅ o i l l e
!                    h a l l i t:d s:z i j:∅ a
!                    j u o k:x s:∅ i j:∅ a n
!                    p a l v:w e l i j:∅ a
!                    p a l v:w e l i j:∅ o i t a
!                    r a n k:g a i s i j:∅ a
!                    v:w a a t i j:∅ a n s a:∅
!                    v:w a l e h t e l i j:∅ a t
!                    v:w a r t i j:∅ a
              s i _ [a oSuff | o oSuff | o i t ] ;
!                    s i j:∅ a
              t e k i _ ä ;
!                    t e k i j:∅ ä

"k:c" k:c => \:k _ [k | (:∅) [:a | :o | :u] | ∅:h | l | r] ;
!                    j a l k:c a i n s a:∅

"kk:ck" k:c <= _ k ;

"k:g" k:g => n _ ;
!                    e n k:g ä
            .#. t y _ ö ;
!                    t y k:g ö s i:∅

"k:x" k:x <=> _ s:∅ ;
!                    h a a k:x s:∅ i

"ll:l∅" l:∅ => l _ ;
!                    e h t o o:∅ l l:∅ i s e n
```

```
"mm:m∅" m:∅ => m _ Vowel: ;
!                    i s ä m m:∅ e:∅

!"nn:n∅" n:∅ => n _ e:∅ .#. ;
!                    k ä s i ä n n:∅ e:∅
!                    t e i t ä n n:∅ e:∅
!                    a j a t u k:x s:∅ i a n n:∅ e:∅
!                    k y m m e n e n n:∅ e n

"o:∅" o:∅ => o: _ ;
!                    e h t o o:∅ n a

"o:a" o:a => k: _ o: n .#. ;
!                    k:c u u l k:c o:a o:∅ n
"~ oo:oa" o:o /<= o:a _ ;

"p:b" p:b => m _ ;
!                    s u u r e m p:b i
!                    ?? s a p:b ∅:b a t h:t i
!                    ?? m u ∅:u l p:b e:ä r i n
!                    ?? t o p:b i a a:∅ n

"p:w" p:w => _ [u | y| ä] :i (s i: (v:∅ ä:∅ t)) .#. ;
!                    v:w i i:j p:W y i
!                    l u o p:w u i
!                    r e p:w ä i s i:∅
!                    l e p:w ä ∅:i s i v:∅ ä:∅ t

"pp:p∅" p:∅ => :Vowel (:m | :l | :r) p _ ;
!                    k:c u m p p:∅ a n i

"s:∅" s:∅ => s _ ;
!                    e d e s s:∅ ä
!                    s e a s s:∅ a:∅
             :x _ ;
!                    h a a k:x s:∅ i

"s:n" s:n => s _ [e | u | y] t .#. ;
!                    n o u:∅ s s:n u t
!                    k:c a t k:c a i s s:n e e:∅ t

"s:z" s:z => t: _ ;
!                    e t:d s:z i
             .#. j o k: a i ∅:d _ e ;
!                    j o k:c a i ∅:d s:z e l l e

"t:d" t:d => [a|e|i|o|u (u:∅)|y|ä|ö|h|l|n|t:] _ [a|e|i|o|u|y|ä|ö] ;
!                    p e l t:d o
             _ s:z ;
```

```
!                   p a i t:d s:z i
            .#. _ u o m [a | i] ;
!                   t:d u o m i o n
"lt:ll" t:l => .#. :Cons* :Vowel (:Vowel) l _ ;
!                   k:c u l t:l a i n e n

"t:n" t:n => n _ [[o|u] i (v: a:∅) (t)] .#. ;
!                   a n t:n o i
!                   i l m a a:∅ n t:n u i
!                   a n t:n o i v:∅ a:∅ t
            n _ [[ö|y] i (v: ä:∅) (t)] .#. ;
!                   s y n t:n y i v:∅ ä:∅ t
            n _ [a|ä|y] i s i (v:∅ [a:∅|ä:∅]) t .#. ;
!                   a n t:n a i s i v:∅ a:∅ t

"rt:rr" t:r => r _ a i s ;
!                   k:c u m a r t:r a i s i:∅

"t:∅" t:∅ => t _ ;
!                   p r o f:p ∅:h e e:∅ t t:∅ a i n

"u:∅" u:∅ => u _ ;
!                   h a l t:d u u:∅ n
!                   p a k:c a n a l l i s u u:∅ d e s t a:∅
            n o _ s s:n ;
!                   n o u:∅ s s:n u t

"v:f" v:f => .#. _ [a n :g | i :c u n a] ;
!                   v:f a n k:g i n a
!                   v:f i k:c u n a

"v:g" v:g => u _ u ;
!                   s u v:g u n
!                   l u v:g u n
!                   r i u v:g u l l a

"v:∅" v:∅ => i _ [a:∅|ä:∅] t .#. ;
!                   s a i s v:∅ a:∅ t

"y:∅" y:∅ => y _ ;
!                   v:w ä ä r y y:∅ t t ä

"ä:∅" ä:∅ => ä _ ;
!                   k ä ä:∅ r m e e:∅ n
            [e | n s | s s:∅ | s t] _ .#. ;
!                   h e t k e ä:∅
!                   n ä k ö n s ä:∅
!                   h e n g e s s:∅ ä:∅
            .#. t i e t _ k [ä ä:∅ ∅:t| ö] ;
!                   t i e t ä:∅ k ä ä:∅ ∅:t
```

```
             i v:Ø _ t .#. ;
!                    k ä ä n t:n i v:Ø ä:Ø t

"ö:Ø" ö:Ø => k ö: _ [n | t] .#. ;
!                    ä l k ö ö:Ø n

"ö:ä" ö:ä => _ ö: n .#. ;
!                    ä l k ö:ä ö:Ø n

"~ öö:äö" ö:ö /<= ö:ä _ ;

"Ø:d" Ø:d => _ s:z ;
!                    j o k:c a i Ø:d s:z e l l e
!                    j o u 0:d s:z e n
!                    ä k k:Ø i Ø:d s:z e l t ä

"Ø:e" Ø:e => .#. [e :d :z | k ä r s | k ä ä r | p y :Ø h k |
                  r u o :c k | s a l | s o t: | v: a a t:] _ i .#. ;
!                    e t:d s:z Ø:e i
!                    k ä r s Ø:e i
!                    k ä ä r Ø:e i
!                    p y y:Ø h k Ø:e i
!                    r u o k:c k Ø:e i
!                    s a l l Ø:e i
!                    s o t:d Ø:e i
!                    s o t Ø:e i
!                    v:w a a t:d Ø:e i

"Ø:g" Ø:g => .#. [a i|a l|j a|p a|t e|k:c o (r)|r u o|h u o|n ä]
             _ [a|e|o|u|y|ö] ;
!                    a i Ø:g o i t
!                    a l Ø:g u s t a
!                    h u o Ø:g a t a
!                    j a Ø:g a t t e
!                    k:c o Ø:g o s s:Ø a:Ø
!                    k:c o Ø:g o l l a
!                    k:k o r Ø:g o t a n
!                    n ä Ø:g y n
!                    n ä Ø:g ö n
!                    p a Ø:g o s t a:Ø
!                    r u o Ø:g o n
!                    t e Ø:g o i l l a
!                    v:w a a ':g a l l a

"Ø:h" Ø:h => f:p _ ;
!                    f:p Ø:h a r i s e u s t e n
             a _ a n .#. ;
!                    j u h l a Ø:h a n
             e _ e n .#. ;
!                    h ä n e Ø:h e n
```

```
              i _ i n .#. ;
!                     k:c a r i Ø:h i n
              o _ o n .#. ;
!                     a r m o Ø:h o n
              u _ u n .#. ;
!                     l o p p u Ø:h u n
              ä _ ä n .#. ;
!                     e l ä m ä Ø:h ä n
              ö _ ö n .#. ;
!                     k i v:w i s t ö Ø:h ö n
              .#. k:c _ r i s t ;
!                     k:c Ø:h r i s t u s

"asi:ais" Ø:i => [a|ä] _ s i:Ø ;
!                     a v:w a Ø:i s i:Ø
                  [a|ä] _ s i v:Ø [a:Ø|ä:Ø] t .#. ;
!                     l e p:w ä Ø:i s i v:Ø ä:Ø t
              k:c u k:c k o _ .#. ;
!                     k:c u k:c k o Ø:i
              o _ n u t ;
!                     a i k:c o Ø:i n u t

"Ø:n" Ø:n => t:d s:z e _ .#. ;
!                     y l i t:d s:z e Ø:n
!                     o h i t:d s:z e Ø:n
!                     l ä p i t:d s:z e Ø:n
!                     e d i t:d s:z e Ø:n
!                     a l a i t:d s:z e Ø:n

"Ø:s" Ø:s => .#. :c a n s _ [:a | :o] ;
!                     k:c a n s Ø:s a n

"Ø:t" Ø:t => k: [a a:Ø | ä ä:Ø] _ .#. ;
!                     a n t:d a k:c a a:Ø Ø:t
              Vowel: Cons:+ Vowel:+ Cons:+ [a|o|i] _ a .#. ;
!                     a s i a Ø:t a
!                     p a h e m p:b i Ø:t a
              Vowel: Cons:+ Vowel:+ Cons:+ [ä|i] _ ä .#. ;
!                     k y y n ä r ä Ø:t ä
"Ø:w" Ø:w => l _ o i [l | s] ;
!                     j a l Ø:w o i l l a
```

Appendix 4: Distances for automatic character by character alignment

The following short Python program builds a WFST which relates MSF word forms to OLF word forms. The resulting WFST is used in the alignment script in Figure 5. The WFST restricts the character by character matching by rejecting most consonant to vowel and vowel to consonant correspondences. Furthermore it gives penalty weights to letter correspondences depending on how many of their features differ and how much they differ. The numerical values used in the program are more or less arbitrary and one may tune them in order to improve the accuracy.

The program was made for written Finnish language, but one could modify it in order to use it for some other languages. In particular, it would be interesting to extend it so that it would cover phonetic IPA representations of any language.

```
"""Produces a kind of a distance matrix between
characters in an alphabet."""
import sys, io
import libhfst
algfile = libhfst.HfstOutputStream(filename="chardist.fst")

vowels = {
    'i':('Close','Front','Unrounded'),
    'y':('Close','Front','Rounded'),
    'u':('Close','Back','Rounded'),
    'e':('Mid','Front','Unrounded'),
    'ö':('Mid','Front','Rounded'),
    'o':('Mid','Back','Rounded'),
    'ä':('Open','Front','Unrounded'),
    'a':('Open','Back','Unrounded')
    }
cmo = {'Close':1, 'Mid':2, 'Open':3}
fb = {'Front':1, 'Back':2}
ur = {'Unrounded':1, 'Rounded':2}

consonants = {
    'm':('Bilab','Voiced','Nasal'),
    'p':('Bilab','Unvoiced','Stop'),
    'b':('Bilab','Voiced','Stop'),
    'v':('Labdent','Voiced','Fricative'),
    'w':('Labdent','Voiced','Fricative'),
    'f':('Labdent','Unvoiced','Fricative'),
    'n':('Alveolar','Voiced','Nasal'),
    't':('Alveolar','Unvoiced','Stop'),
    'd':('Alveolar','Voiced','Stop'),
    's':('Alveolar','Unvoiced','Sibilant'),
    'l':('Alveolar','Voiced','Lateral'),
    'r':('Alveolar','Voiced','Tremulant'),
    'j':('Velar','Voiced','Approximant'),
    'k':('Velar','Unvoiced','Stop'),
    'g':('Velar','Voiced','Stop'),
    'h':('Glottal','Unvoiced','Fricative')}
pos = {'Bilab':1, 'Labdent':1, 'Alveolar':2, 'Velar':3, 'Glottal':4}
voic = {'Unvoiced':1, 'Voiced':2}
def cmodist(x1, x2):
    """Computes a distance of Close/Mid/Open and returns it"""
    return abs(cmo[x2] - cmo[x1])
```

```
def posdist(x1, x2):
    """Computes a distance of articulation position and returns it"""
    return abs(pos[x2] - pos[x1])

def adist(x1, x2):
    """Computes a distance between symbols"""
    return (0 if x1 == x2 else 1)

def printlset(lset):
    """Print the set of letters and their features"""
    ll = sorted(lset.keys());
    flist = []
    for l in ll:
        (x,y,z) = lset[l]
        flist.append("{} : {},{},{}".format(l, x, y, z))
    print('\n'.join(flist))

def featmetr(lset1, lset2, f1, f2, f3):
    """Compute all metric distances between letters in d1 and d2
 according to their features."""
    ll1 = sorted(lset1.keys())
    ll2 = sorted(lset2.keys())
    ml = []
    for l1 in ll1:
        (x1,y1,z1) = lset1[l1]
        for l2 in ll2:
            (x2,y2,z2) = lset2[l2]
            dist = f1(x1,x2) + f2(y1,y2) + f3(z1,z2)
            ml.append("{}:{}::{}".format(l1,l2,dist))
    return (ml)

vvlist = featmetr(vowels, vowels, cmodist, adist, adist)
cclist = featmetr(consonants, consonants, posdist, adist, adist)
vowl = sorted(vowels.keys())
cons = sorted(consonants.keys())
letters = sorted(vowl + cons)
dellist = ['{}:Ø::{}'.format(l,3) for l in letters]
epelist = ['Ø:{}::{}'.format(l,3) for l in letters]
dbllist = ['{} Ø:{}::{}'.format(l,l,2) for l in letters]
sholist = ['{} {}:Ø::{}'.format(l,l,2) for l in letters]

speclist = ['k:c::0 k::0', 'k:x s:Ø::0', 't:d s:z::0', 'Ø:d s:z::3',
            'i:j::1', 'j:i::1', 'i j:Ø::0', 'i i:j::0',
            'f:p Ø:h::0', 'u:v::1', 'v:u::1', 'u:w::1', 'k:c::1',
            '[o:Ø o:?]::5']
all = vvlist + cclist + dbllist +
      sholist + dellist + epelist + speclist
re = '[{}]*'.format(' | '.join(all))

algfst = libhfst.regex(re)
```

```
algfile.write(algfst)
algfile.flush()
algfile.close()
```

References

Beesley, K. R. and L. Karttunen. 2003. Two-level rule compiler. https://web.stanford.edu/~laurik/.book2software/

Biblia. 1642. Biblia. Se on: Coco Pyhä Ramattu, Suomexi. Päramattuin, Hebrean ja Grecan jälke: Esipuhetten, Marginaliain, Concordantiain, Selitösten ja Registerein cansa. Pipping 42, Henrik Keyser, Stockholmis.

Borin, L., M. Forsberg and J. Roxendal. 2012. Korp – the corpus infrastructure of Språkbanken. In Proceedings of LREC 2012. Istanbul: ELRA, 474–478.

Grönros, E.-R. and Kotimaisten kielten tutkimuskeskus. 2006. Kielitoimiston sanakirja. Helsinki: Kotimaisten kielten tutkimuskeskus. http://www.kielitoimistonsanakirja.fi/

Karttunen, L., K. Koskenniemi and R. M. Kaplan. 1987. A compiler for two-level phonological rules. In M. Dalrymple, R. Kaplan, L. Karttunen, K. Koskenniemi, S. Shaio and M. Wescoat (eds.) Tools for morphological analysis. Palo Alto, CA: Center for the Study of Language and Information, Stanford University. 1–61.

Koskenniemi, K. 2013a. Finite-state relations between two historically closely related languages. In Proceedings of the Workshop on Computational Historical Linguistics at NODALIDA 2013. Linköping: Linköping University Electronic Press, 53–53.

Koskenniemi, K. 2013b. An informal discovery procedure for two-level rules. Journal of Language Modelling 1. 155–188.

Koskenniemi, K. 2017. Aligning phonemes using finte-state methods. In Proceedings of the 21st Nordic Conference on Computational Linguistics. Gothenburg: Association for Computational Linguistics, 56–64.

Lindén, K., E. Axelson, S. Hardwick, T. A. Pirinen and M. Silfverberg. 2011. HFST – Framework for compiling and applying morphologies. In C. Mahlow and M. Piotrowski (eds.) Systems and Frameworks for Computational Morphology 2011 (SFCM-2011). Berlin & New York: Springer. 67–85.

Pirinen, T. A. 2015. Omorfi – Free and open source morphological lexical database for Finnish. In Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015). Vilnius,: Linköping University Electronic Press, 313–315.

Sadeniemi, M. (ed.). 1951–1961. Nykysuomen sanakirja. Porvoo & Helsinki: WSOY.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 353–368

Independently generated languages

MARCUS KRACHT
Bielefeld University
marcus.kracht@uni-bielefeld.de

TOMASZ KOWALSKI
La Trobe University, Melbourne
t.kowalski@latrobe.edu.au

KEYWORDS

compositionality
autonomy
independence
independently generated language

ABSTRACT

A language is compositional if the meaning of an expression is a function of the meanings of its parts, determined by the mode of composition. A dual syntactic property is known as the autonomy of syntax: the form of an expression is independent of the meanings of its parts. It is easy for a language to satisfy the property on one side, as long as the other side is completely unconstrained. Satisfying both of them simultaneously is much harder. We call this property independence, and investigate conditions under which a language is independently generated. We conjecture that there exist non-independently generated languages.

1. Compositionality versus Independence

How do humans understand the meaning of a complex sentence they have never heard before? The answer is that there is an algorithm that allows to compute the meaning of the complex expression from its parts. While this much seems uncontroversial, semanticists have actually argued that natural languages possess a stronger property, that of compositionality. A language is said to be compositional if the meaning of a complex expression is a function of the meaning of its parts given the mode of composition; thus, a language is compositional if the algorithm computing the meaning can do so without knowing the expressions that carry these meanings. It is this latter property that has been made into a litmus test for formal semantic theories. A theory that provides a compositional account of meaning is preferred over one that does not. But how much of a constraint is compositionality on a language? In other words, what empirical significance does it have to say that a language is compositional? Do noncompositional languages at all exist?

While the introduction of the subject is often credited to Montague, it is perhaps the work that has been done in the wake of Montague that had put compositionality firmly on the agenda of formal semantic theorising, see Janssen (1997) for an account by one of the protagonists. The survey books Barker & Jacobson (2007) and Hinzen et al. (2012) document the persistent interest in this notion. From a mathematical point of view, the question is how much of an empirical content this notion has. Janssen has actually shown that any language is compositional, provided no constraints on syntactic operations are being made (see Janssen 1997). Though his notion of language is slightly nonstandard, the result holds also for languages in the Saussurean sense, i.e., relations between expressions and meanings.

However, as Kracht (2011) has pointed out, the dual property, namely that the *form* of an expression is independent of the meaning of its parts, is actually a well known hypothesis of generative grammar: It is called the *autonomy of syntax*. It is curious that to our knowledge no definition of this notion with the same explicitness has ever been given in print. In retrospect it seems that a grammar that satisfies both of them simultaneously is really what linguistics have been after, and not compositionality alone. We call this property *independence*. It has a rather simple mathematical formulation, so the investigation may also be of interest in combinatorial theory.

Moreover, it turns out that the property of independence is actually rather tricky. It is still unclear whether there exists a countable language that is not independently generated. Though we believe that such a language exists, we have not been able to find one. The results here exhibit some positive results (specifying languages that are independently generated) and reduces the complexity of the original problem somewhat.

We thank our reviewer for suggesting ways to improve this paper. Also, Marcus Kracht wishes to thank András Kornai for his friendship and the endless discussions on mathematics, language and life.

2. Autonomy and compositionality

A *language* is an arbitrary subset L of $E \times M$, where E and M are given sets of expressions and meanings, respectively. An *independent grammar* for L is a finite set $F \subseteq L$ and a finite set P of pairs of functions (f_i, g_i),

$i < m$, such that for $i < m$ there is an n_i (the arity of the functions) such that $f_i : E^{n_i} \hookrightarrow E$ and $g_i : M^{n_i} \hookrightarrow M$ are both partial functions and L is exactly the set that can be generated from F using P. The action of such a pair is defined as usual: $(f, g)((e_0, m_0), \cdots, (e_{n_i-1}, m_{n_i-1}))$ is defined if and only if both $f(e_0, \cdots, e_{n_i-1})$ and $g(m_0, \cdots, e_{n_i-1})$ are defined and in that case

$$(f, g)((e_0, m_0), \cdots, (e_{n_i-1}, m_{n_i-1})) := (f(e_0, \cdots, e_{n_i-1}), g(m_0, \cdots, m_{n_i-1}))$$

There is an obvious mathematical generalisation. Let $R \subseteq \omega^d$ be a relation. Say that R is *independently generated* if there is a finite set of partial functions f_i^k, $k < d$, of arity n_i (only dependending on i) such that the product functions $(f_i^0, f_i^1, \cdots, f_i^{d-1})$ (of arity n_i) generate R from a finite subset. The limiting case of $d = 1$ is trivial. Any countable subset of ω can be generated; it suffices to pick one constant and a single unary function. Thus, the case $d = 2$ is the first really interesting case. Notice also that if there is a relation of arity d that is not independently generated, then there is an example in any higher arity.

Notation. If f is a function and S a set, put $f[S] := \{f(x) : x \in S\}$. Given $L \subseteq \omega^2$ we write L_i for the “column” $\{j : (i, j) \in L\}$. It follows that $L = \bigcup_{i \in \omega} \{i\} \times L_i$. Dually we write ${}^jL := \{i \in \omega : (i, j) \in L\}$.

The notion of independence is restrictive. Let $L \subseteq E \times M$ be countably infinite. Then there exists a finite subset F and a finite number of partial functions on $E \times M$ (rather than independent functions on E and M) generating L. Indeed, one constant plus a single unary function is enough. Simply enumerate $L = \{(e_i, m_i) : i \in \omega\}$, put $F := \{(e_0, m_0)\}$ and let $f : E \times M \to E \times M$ be defined by

$$f((e, m)) := \begin{cases} (e_{i+1}, m_{i+1}) & \text{if } (e, m) = (e_i, m_i) \\ (e, m) & \text{else} \end{cases}$$

By assumption, i is unique in the first case. Then it is easily seen that $(e_k, m_k) = f^k((e_0, m_0))$, so we generate exactly L.

However, the question was whether it is possible to define the functions in such a way that the actions on E and on M are independent of each other. In the linguistic literature, two weaker notions have been discussed. The most important one is compositionality. The first result will be that all countable languages have a compositional grammar.

Since the set of generating functions is finite, L is at most countably infinite. Thus we can restrict E and M to some countably infinite subset; without loss of generality we can assume them to be the set of natural numbers $\omega = \{0, 1, 2, 3, \cdots\}$. (Formally, there is nothing that distinguishes members of E from members of M.) Thus, from now on $E = M = \omega$. Let $p_1 : \omega^2 \to \omega : (i,j) \mapsto i$ and $p_2 : \omega^2 \to \omega : (i,j) \mapsto j$.

Definition 1. *Let f be a partial n-ary function on ω^2. We say f is* ***independent in the first component*** *if for all pairs $(i_0, j_0), \cdots, (i_{n-1}, j_{n-1})$ and all pairs $(i_0, k_0), \cdots, (i_{n-1}, k_{n-1})$: $f((i_0, j_0), \cdots, (i_{n-1}, j_{n-1}))$ is defined if and only if $f((i_0, k_0), \cdots, (i_{n-1}, k_{n-1}))$ is defined, and if any of them is defined, then the first projection of the values are identical, i.e.,*

$$p_1(f((i_0, j_0), \cdots, (i_{n-1}, j_{n-1}))) = p_1(f((i_0, k_0), \cdots, (i_{n-1}, k_{n-1}))).$$

Dually the notion of ***independence in the second component*** *is defined.*

Theorem 2. *Let $L \subseteq \omega^2$. Then there is a finite set of functions generating L from a finite subset where all functions are independent in the first component. Likewise, the is a finite set of functions generating L from a finite subset where all functions are independent in the second component.*

Proof. Obviously, we need to prove only the first claim. The second follows analogously. (Or instead, apply the method to $L^{\smile} := \{(j,i) : (i,j) \in L\}$ and then "switch" the solution.) Now, consider first the language

$$M := \{(i,j) : L_i \neq \varnothing, j \text{ is minimal in } L_i\}.$$

Clearly, $M \subseteq L$.

So M has the form $M = \{(i, n_i) : i \in H\}$ for some $H \subseteq \omega$. Let α be the least member of H, i.e., the least number such that $L_\alpha \neq \varnothing$. Introduce a constant for (α, n_α). Next, let $k(i,j)$ be defined as follows. In case $j = n_i$ and there is a $q > i$ such that $L_q \neq \varnothing$, let $k(i,j) := (p, n_p)$, where p is the smallest number $> i$ such that L_p is nonempty. If no such number exists, or if $j \neq n_i$, put $k(i,j) := (i,j)$. Now put

$$d((i,j)) := (p, k(i,j))$$

This defines our first function. (A) d is independent in the first component since p can be established from i alone. (B) L is closed under d. For given $(i,j) \in L$, if $d((i,j)) = (i,j)$ then obviously $d((i,j)) \in L$. If $d((i,j)) \neq (i,j)$, then $(i,j) = (i, n_i)$ and $d((i,j))$ therefore has the form (p, n_p), where by definition $(p, n_p) \in M$. (C) M is the closure of $\{(\alpha, n_\alpha)\}$ under d. We prove by induction on i that all $(i, n_i) \in M$ can be generated. For $i = \alpha$ this is the case by assumption. Let i be given with $(i, n_i) \in M$. Then let $p < i$ be the largest number such that $(p, n_p) \in M$. By inductive hypothesis, (p, n_p) is generated from $\{(\alpha, n_\alpha)\}$. But $(i, n_i) = d((p, n_p))$, so it is also generated from $\{(\alpha, n_\alpha)\}$.

Next, define a function ν as follows. Given i and j, $\nu(i,j) := j$ if either $j \notin L_i$ or j is the largest member of L_i. Otherwise, $\nu(i,j)$ yields the least j' such that $j' \in L_i$ and $j' > j$. Now put

$$u((i,j)) := (i, \nu(i,j))$$

(A) u is evidently independent in the first component. (B) L is closed under u. For if $(i,j) \in L$ and $u((i,j)) = (i,j)$ then $u((i,j)) \in L$. Otherwise, $u((i,j)) = (i, \nu(i,j)) = (i,j')$, where among other $j' \in L_i$. So, $u((i,j)) \in L_i \subseteq L$. (C) L is generated from M using u. This is proved by induction on j. Choose $(i,j) \in L$. If j is minimal in L_i then $j = n_i$ and the claim trivially follows. Otherwise, choose j' to be maximal such that $(i,j') \in L$ and $j' < j$. By inductive hypothesis, (i,j') is generated from M using u. But $u((i,j')) = (i,j)$, and so (i,j) is likewise generated from M using u. □

Notice that we have been able to define total functions. Consider a system F of generating (partial) functions on $E \times M$. This system is called **compositional** if each member $f \in F$ is independent in the second component; it is called **autonomous** if each member $f \in F$ is independent in its first component. We can rephrase the previous theorem as follows. A language is **compositional** (**autonomous**) if it has a finite compositional (autonomous) set of generating functions.

Corollary 3. *Let L be a countable language.*

- *L is autonomous.*
- *L is compositional.*

□

Here is a surprising consequence.

Corollary 4. *Suppose that L is either many-to-one (= unambiguous) or L is one-to-many. Then L is independently generated.*

Proof. Consider the second case, i.e., assume that L is one-to-many (the other case being dual). Let $f : M \to E$ be such that $f(j)$ is the unique i such that $(i,j) \in L$. Define the grammar as in the proof of Theorem 2. Now put

$$d((i,j)) := (p, k(f(j), j))$$

as well as

$$u((i,j)) := (i, \nu(f(j), j))$$

where k and ν are defined as before. By assumption, $i = f(j)$, so that this defines the same function. The so-defined functions do not depend on the first component any more, and so are independent. $\square$

Corollary 5. *Let $L \subseteq E \times M$ a language such that for some $A \subseteq E$, $B \subseteq M$, $L \cap A \times B$ is a many-to-one or one-to-many relation on $A \times B$ containing an infinite partial bijection. Then L is independently generated.*

Proof. We generate $L \cap A \times B$ by means of independent partial functions defined on $A \times B$, as shown in Corollary 4. L contains a partial infinite bijection $\{(a_i, b_i) : i < \omega\}$. Let $L - A \times B = \{(e_i, m_i) : i < \omega\}$. Now add a new unary partial function $f : A \times B \to (E - A) \times (M - B)$ with

$$f((x,y)) = \begin{cases} (e_i, m_i) & \text{if } (x,y) = (a_i, b_i) \\ \text{undefined} & \text{else} \end{cases}$$

By assumption, i is uniquely determined by x alone *and* by y alone, so f is actually independent. $\square$

The notion of independence for languages is not the conjunction of autonomy and compositionality (if it were, all languages would be independent, by Corollary 3); indeed, it is much stronger than that. For it says that the language has an independent grammar, that is, a grammar where every function is independent in both components. This is what we are going to study now.

3. Basic results

In using partial functions, here is a trick that will be used on several occasions. Denote by $[F]_P$ the closure of F under P. Let A be the disjoint union of B and C. Let P be a set of partial functions on B, and Q a set of partial function on C. Take $B_0 \subseteq B$ and $C_0 \subseteq C$. Then $[B_0 \cup C_0]_{P \cup Q} = [B_0]_P \cup [C_0]_Q$. To see that notice that functions from P are undefined on every tuple containing elements from C, and functions from Q are undefined on every tuple containing an element from B. Therefore, functions from P cannot act on outputs of functions from Q, and vice versa.

Lemma 6. *Let $L \subseteq \omega^2$, and $\omega = E' \cup E''$, with E' and E'' disjoint, and let $M \subseteq \omega$. Now put $L' := L \cap E' \times M$, $L'' := L \cap E'' \times M$. Then if both L' and L'' are independently generated, so is $L \cap \omega \times M$.*

Indeed, simply take the (disjoint) union of the constants and functions. The following two claims are obvious.

Lemma 7. *If L is independently generated, so is $L^{\smile} := \{(j, i) : (i, j) \in L\}$.*

Lemma 8. *Let $\pi, \rho : \omega \to \omega$ be injections. Let $(\pi, \rho)[L] := \{(\pi(e), \rho(m)) : (e, m) \in L\}$. Then $(\pi, \rho)[L]$ is independently generated iff L is independently generated.*

There is a special corollary of this theorem that is worth stating separately. Consider the case where $L_i = \varnothing$ for certain i. Denote by $U := \{i : L_i \neq \varnothing\}$. If U is infinite there is a bijection $\nu : U \to \omega$. Consider the language $L^\bullet := (\nu, id_M)[L]$. We have $(L^\bullet)_i = L_{\nu(i)} \neq \varnothing$ for all $i \in \omega$.

Corollary 9. *$L^\bullet$ is independently generated iff L is.*

If U is finite, L is independently generated anyway, by the next theorem.

Lemma 10. *Let n be a finite number.*

1. *Every finite language is independently generated.*
2. *$n \times n$ is independently generated.*
3. *$\omega \times \omega$ is independently generated.*
4. *$\omega \times n$, $n \times \omega$ are independently generated.*
5. *Every cofinite subset of $\omega \times \omega$ is independently generated.*

Proof. The first claim is easy. Just introduce a constant for every element of L. The second claim follows immediately. To show the third claim introduce a constant for $(0, 0)$, and two unary functions: one sending (i, j) to $(i + 1, j)$, and one sending (i, j) to $(i, j + 1)$. The fourth claim is proved thus. For each $j < n$ take a constant for $(0, j)$. Finally, add a single unary function sending (i, j) to $(i + 1, j)$. For the last claim, let $L = \omega \times \omega - \{(i_k, j_k) : k < n\}$. Put $E_0 := \{i_k : k < n\}$, $E_1 := \omega - E_0$; $M_0 := \{j_k : j < n\}$, $M_1 := \omega - M_0$. Now $L \cap E_0 \times M_0$ is finite; $L \cap E_0 \times M_1 = E_0 \times M_1$, $L \cap E_1 \times M_0 = E_1 \times M_0$, and $L \cap E_1 \times M_1 = E_1 \times M_1$. The first is independently generated since it is finite. The others are independently generated because they are a simple product of at most countable sets. Now use Lemma 6. ☐

Say that L is *essentially bounded* if $L \subseteq n \times \omega$ or $L \subseteq \omega \times n$.

Lemma 11. *Every essentially bounded language is independently generated.*

Proof. >From Lemma 10 by repeated application of Lemma 6. ☐

Next we are going to reduce the problem even further. Let $H \subseteq \omega$ such that for every $i \in H$ there is a $j \notin H$ and $L_j = L_i$. Then put

$$L^{-H} := \{(i,j) : (i,j) \in L, i \notin H\}$$

Lemma 12. *If L^{-H} is independently generated then L is independently generated.*

Proof. Suppose that L^{-H} is independently generated. For $j \notin H$ put

$$B_j := \{k : k \in H, L_k = L_j\}$$

Now let $h : \omega \to \omega$ be defined as follows. If $j \notin H$ and $B_j = \varnothing$ then $h(j) := j$. If $j \notin H$ and $B_j \neq \varnothing$, then let $h(j) := \min B_j$. Else, if $j \in H$ then $j \in B_i$ for some i. If $j = \max B_i$, put $h(j) := j$, otherwise let $h(j)$ be the least $j' \in B_i$ such that $j' > j$. Finally, let f be defined by

$$f((i,j)) := (h(i), j)$$

(A) f is independent. (B) L is closed under f. Consider $(i,j) \in L$. If $f((i,j)) = (i,j)$ then $f((i,j)) \in L$. Otherwise, $f((i,j)) = (h(i),j)$, where by definition $L_{h(i)} = L_i$. Thus, $(h(i),j) \in L_{h(i)}$ and so $(h(i),j) \in L$. (C) L is generated from L^{-H} using f. If not, let i be minimal such that for some j, $(i,j) \in L$ but it is not generated from L^{-H} using f. Then $i \in H$. Let i' the largest number such that $i' < i$ and $i' \in H$ if it exists, else let $i' \notin H$ such that $i = \min B_{i'}$. By definition, $i = h(i')$. It is easily seen that (i',j) is generated from L^{-H} using f; the same is now true for (i,j). ☐

Thus we can restrict our search for nonindependent languages to those subsets of ω^2 where all columns are different and all rows are different.

4. Main theorems

We are going to investigate three conditions under which languages are independently generated. The second and third conditions both generalise the first, in slightly different directions. Examples will show that the generalisations are proper.

Definition 13. *Let $L \subseteq \omega \times \omega$. Say that L is **n-discriminable** if there is a family $\{A_i : i \in \omega\}$ of sets such that:*

1. *for each i: $0 < |A_i| \leq n$;*
2. *for each i, j: if $i \neq j$ then $A_i \neq A_j$;*

3. *for every i:* $A_i \subseteq L_i$*; and*

4. *for every* i, j $A_j \subseteq L_i$ *if and only if* $j = i$.

In that case, we call the family $\{A_i : i \in \omega\}$ *an* ***n-discriminating family*** *for L. (Notice that 4. implies 3.)*

Notice that by definition, $A_i \nsubseteq A_j$ for $i \neq j$. For if $A_i \subseteq A_j$ we have $A_i \subseteq A_j \subseteq L_j$, from which by definition $i = j$.

Theorem 14. *Let L be n-discriminable. Then L is independently generated.*

Proof. Let $\{A_i : i \in \omega\}$ be a discriminating family for L. Let $\overline{A}_i$ be a sequence of length n that enumerates A_i, possibly repeating an element to reach length n. (Eg if $n = 4$ and $A_2 = \{3, 6, 7\}$ then $\overline{A}_2 = \langle 3, 6, 7, 7\rangle$ is a possible choice.) For each member of $\{(0, i) : i \in A_0\}$ we introduce a constant. Now we define the following n-ary functions f^k, $k < n$. Let h be the kth member of $\overline{A}_{i+1}$.

$$f^k((i, j_0), (i, j_1), \cdots, (i, j_{n-1})) := \begin{cases} (i+1, h) & \text{if } \langle j_0, j_1, \cdots, j_{n-1}\rangle = \overline{A}_i \\ \text{undefined} & \text{else} \end{cases} \tag{1}$$

Clearly this function is independent: on the first component it gives $i+1$ if all arguments are identical to i and is undefined otherwise. On the second component it gives h if the arguments are exactly given as in $\overline{A}_i$, and is undefined else. The partiality seems to be essential here.

Now define a single $n+1$-ary function g with the following action. For each i we assume $L_i - A_i$ to be enumerated as $\{k^i_j : j < \kappa_i\}$ where $\kappa_i < \omega + 1$ (so κ_i can be finite or $= \omega$).

$$g((i, j_0), (i, j_1), \cdots, (i, j_n)) := \begin{cases} (i, k^i_0) & \text{if } \langle j_0, j_1, \cdots, j_{n-1}\rangle = \overline{A}_i \\ & \quad \text{and } j_n = j_{n-1},\ \kappa_i \neq 0 \\ (i, k^i_{p+1}) & \text{if } \langle j_0, j_1, \cdots, j_{n-1}\rangle = \overline{A}_i \\ & \quad \text{and } j_n = k^i_p,\ p+1 < \kappa_i \\ \text{undefined} & \text{else} \end{cases} \tag{2}$$

So defined g is independent. On the first coordinate the output is i if all inputs equal i; and is undefined elsewhere. On the second coordinate it yields the next element in the enumeration if there is one (and repeats the element if it is the last in the enumeration), provided the first n arguments equal $\overline{A}_i$; and is undefined elsewhere.

It now remains to be shown that this set of functions generates exactly L. There are two parts: (i) the functions generate all of L; (ii) L is closed under the functions.

To prove (i), we shall first show that all $\{i\} \times A_i$ are generated using the f^k and the constants. Recall that all members of $\{0\} \times A_0$ are values of some constant. Now by induction assume that $\{i\} \times A_i$ is generated. Thus all pairs (i, j_p) exist, $p < n$, where $j_p \in A_i$. Then, using the functions f^k, we can generate $(i + 1, h)$, where h is the kth member of $\overline{A}_{i+1}$. Since all elements of A_{i+1} appear at least once in $\overline{A}_{i+1}$, all of $\{i + 1\} \times A_{i+1}$ is thus generated. Next we show that for every i, $\{i\} \times L_i$ is generated from $\{i\} \times A_i$ using the function g. To that end, recall that $L_i - A_i$ is enumerated as k^i_0, k^i_1 and so on for indices in κ_i. If $\kappa_i = 0$, nothing needs to be done. If $\kappa_i > 0$, we get (i, k^i_0) as the value of $g((i, j_0), (i, j_1), \cdots, (i, j_{n-1}), (i, j_{n-1}))$, and (i, k^i_{p+1}) as the value of $g((i, j_0), (i, j_1), \cdots, (i, j_{n-1}), (i, k^i_p))$. By induction, all values are generated.

Finally, we need to show that L is closed under the functions. Consider

$$f^k((i_0, j_0), (i_1, j_1), \cdots, (i_{n-1}, j_{n-1}))$$

This is defined only if $i := i_0 = i_1 = \cdots = i_{n-1}$ and $\langle j_0, j_1, \cdots, j_{n-1}\rangle = \overline{A}_q$ for some q. We have $q = i$, since $\{i\} \times A_p \subseteq L$ only if $p = i$ by definition of n-discrimination. So, the function is defined only on $f^k((i, j_0), (i, j_1), \cdots, (i, j_{n-1}))$, where $\langle j_0, \cdots, j_{n-1}\rangle = A_i$, and yields the value $(i + 1, j'_k)$, where j'_k is the kth member of $\overline{A}_{i+1}$. By definition, this is in L. Next, consider

$$g((i_0, j_0), \cdots, (i_n, j_n))$$

This is defined only if $i := i_0 = i_1 = \cdots = i_n$. Additionally, like in the case of f^k, the sequence $\langle j_0, j_1, \cdots, j_{n-1}\rangle$ must equal $\overline{A}_i$. Hence we have to look at

$$g((i, j_0), (i, j_1), \cdots, (i, j_n))$$

Several cases arise. (a) $j_n = j_{n-1}$. In that case we get (i, k^i_0), provided that $\kappa_i > 0$. In that case, $L_i - A_i$ is nonempty, and contains k^i_0 by definition. (b) $j_n = k^i_p$, where k^i_p is a member of $L_i - A_i$. In fact, it is the pth member of the enumeration. If $p + 1 = \kappa_i$, then $L_i - A_i$ is exhausted, and g is undefined. If not, we get (i, k^i_{p+1}), which is in $L_i - A_i$ by definition. (c) The function is undefined on all other inputs. In all cases, we get values in L. The proof is complete. □

Corollary 15. *Suppose that there exists an n such that for all $i \in \omega$ $|L_i| \leq n$. Then L is independently generated.*

Proof. By Lemma 12 we can reduce this to the case where for $i \neq j$ $L_i \neq L_j$. Define $I(j) := \{i : |L_i| = j\}$ and $L^j := \cup_{i \in I(j)} L_i$. By Lemma 6, we need to show only that each of the L^j is independently generated. To this end, it is enough to show that $\{L_i^j : i \in I(j)\}$ is a j-discriminating family. This is easy to verify. $\square$

As an application, consider the language $L = \{(i,i),(i,i^2) : i \in \omega\}$. Here we can simply take $A_i := L_i$. Indeed, this is a 2-discriminating family. For $|A_i| \leq 2$, the sets are nonempty, pairwise distinct ($\{i,i^2\} = \{j,j^2\}$ iff $i = j$), and, finally, if $\{i,i^2\} \subseteq L_j$ then $j = i$; for if $\{i,i^2\} = \{j,j^2\}$ then either the sets contain both two members, and then since $i < i^2$, $j < j^2$ we have $i = j$; or they contain one member and then have the form $\{i\} = \{j\}$, from which again $i = j$. So, by the previous result the language is independently generated.

A more complex example, to which this result cannot be applied, though, is $\{(i,i^k) : i,k \in \omega\}$. It is a consequence of the next theorem that this language is independently generated.

Definition 16. *Call a language **weakly n-discriminable** if there is a family $\{A_i : i \in \omega\}$ of sets such that*

1. *for every i, $0 < |A_i| \leq n$;*
2. *for every i,j: if $i \neq j$ then $A_i \neq A_j$; and*
3. *for every i, $A_i \subseteq L_i$; and*
4. *for every i,j: if $A_j \subseteq L_i$ then $L_i \subseteq L_j$. (This is trivially true if $i = j$.)*

In particular, if $A_i \subseteq A_j$ then we must have $L_j \subseteq L_i$. Clearly, all n-discriminable languages are also weakly n-discriminable, but the converse does not hold, as the example just given shows. For if L is n-discriminable, we must have $L_i \not\subseteq L_j$ for all $i \neq j$. (For if $i \neq j$ and $L_i \subseteq L_j$, then since $A_i \subseteq L_i$ we also have $A_i \subseteq L_j$, which is excluded.) But the language $\{(i,i^k) : i,k \in \omega\}$ fails this: we have $L_2 \subseteq L_4$. On the other hand, the family defined by $A_i := \{i,i^2\}$ is a weakly discriminating family. For if $A_i \subseteq L_j$ then $i = j^p$ for some p, whence $L_i = \{i^k : k \in \omega\} \subseteq \{j^p : p \in \omega\}$. The next theorem establishes that this language is independently generated.

Theorem 17. *Let L be weakly n-discriminable. Then L is independently generated.*

Proof. Let $M := \{i :$ *for no* $j < i$: $A_j = A_i\}$. Furthermore, let $B(i) = \{j : A_i = A_j\}$. Thus, M consists of all minimal members of the sets $B(i)$. Now let m and n be unary partial functions with the following action. $m(j)$ is undefined if j is maximal in M, and otherwise it is $m(j) := \min\{k : k \in M, j < k\}$. $n(j)$ is undefined if j is maximal in $B(j)$, and $n(j) := \min\{k : k \in B(j), k > j\}$ otherwise.

We need three sets of functions in addition to constants for the members of $\{0\} \times A_0$. The first contains the functions f^k, $k < n$. Define the sequences $\overline{A}_i$ as above, with the exception that we require $\overline{A}_i = \overline{A}_j$ if $B(i) = B(j)$ (that is, if $A_i = A_j$).

$$f^k((i, j_0), (i, j_1), \cdots, (i, j_{n-1})) := \begin{cases} (m(i), h) & \text{if } i \in M,\ \langle j_0, j_1, \cdots, j_{n-1}\rangle = \overline{A}_i \\ \text{undefined} & \text{else} \end{cases} \tag{3}$$

The second set consists of the h^k, $k < n$.

$$h^k((i, j_0), (i, j_1), \cdots, (i, j_{n-1})) := \begin{cases} (n(i), i_k) & \langle j_0, j_1, \cdots, j_{n-1}\rangle = \overline{A}_i \\ \text{undefined} & \text{else} \end{cases} \tag{4}$$

Finally, we define the function g as above:

$$g((i, j_0), (i, j_1), \cdots, (i, j_n)) := \begin{cases} (i, k_0^i) & \text{if } \langle j_0, j_1, \cdots, j_{n-1}\rangle = \overline{A}_i \\ & \quad \text{and } j_n = j_{n-1},\ \kappa_i \neq 0 \\ (i, k_{p+1}^i) & \text{if } \langle j_0, j_1, \cdots, j_{n-1}\rangle = \overline{A}_i \\ & \quad \text{and } j_n = k_p^i,\ p+1 < \kappa_i \\ \text{undefined} & \text{else} \end{cases} \tag{5}$$

These functions are independent.

Again, we need to show that (i) L is generated from the functions, and (ii) L is closed under these functions. As for (i), we note that by definition, we can generate all A_i where $i \in M$ from $\{0\} \times A_0$. Next, we can generate the $\{j\} \times A_j$ for all $j \in B(j)$ just by applying the h^k, since we have generated its minimal members. Third, by using g we generate the columns L_i.

Now we show that L is also closed under the functions. Consider

$$f^k((i_0, j_0), (i_1, j_1), \cdots, (i_{n-1}, j_{n-1}))$$

This is defined only if $i := i_0 = i_1 = \cdots = i_{n-1} \in M$ and $\langle j_0, j_1, \cdots, j_{n-1}\rangle = \overline{A}_q$ for some q. By definition of M, for two numbers $p, q \in M$, $A_p \neq A_q$, and so $q = i$. The value $(m(i), h)$ is in $A_{m(i)}$ by definition of f^k. Next consider

$$h^k((i_0, j_0), (i_1, j_1), \cdots, (i_{n-1}, j_{n-1}))$$

This is defined only if $i := i_0 = i_1 = \cdots = i_{n-1}$, and $\langle j_0, j_1, \cdots, j_{n-1}\rangle = \overline{A}_q$ for some q. The value is $(n(i), j_k)$; while j_k is again in $\overline{A}_q$, the new index is $n(i)$. However, by choice of the function n, $A_{n(i)} = A_i$, so we get a value from $A_{n(q)}$. Thus, these functions are only defined on $\bigcup_q \{q\} \times A_q$ and yield values in that set.

Finally, we need to show that L is closed under g. Consider

$$g((i_0, j_0), \cdots, (i_n, j_n))$$

This is defined only if $i := i_0 = i_1 = \cdots = i_n$ and $\langle j_0, \cdots, j_{n-1} \rangle = \overline{A}_q$ for some q. Now, suppose that $A_q = \{j_0, \cdots, j_{n-1}\} \subseteq L_i$. Then by assumption on weak discriminability, $L_q \subseteq L_i$. Hence, two cases arise. (i) $q = i$. Then by definition of g, the value is in L_i. (ii) $p \neq i$. Then, since the value is in L_q, and $L_q \subseteq L_i$, it is also in L_i. □

Definition 18. *Call L* ***boundedly discriminable*** *if there are numbers n and n', an infinite set $M \subseteq \omega$ and a family $\{A_i : i \in M\}$ of sets such that the following holds:*

1. *for each $i \in M$, $|A_i| \leq n$;*
2. *for each $i \in M$: $A_i \subseteq L_i$;*
3. *for each $i \in M$, the set $B(i) := \{j : A_i \subseteq L_j\}$ has at most n' elements; and*
4. *for each $i, j \in M$, $i \neq j$, $B(i) \cap B(j) = \varnothing$.*

Every n-discriminable language is boundedly discriminable; just take $M := \omega$. The sets $B(i)$ each have only one member in this case, so $n' := 1$.

Actually, it follows that for each $i, j \in M$, $i \neq j$, $A_i \neq A_j$. For if $i, j \in M$ and $i \neq j$, the last clause implies $j \notin B(i)$, that is, $A_j \nsubseteq L_j$, so that $A_j \neq A_i$, since $A_i \subseteq L_i$.

Notice that allowing M to be finite would not gain anything, as then the set of indices would be finite, bounded by some multiple of $|M|$. So the only remaining interesting case is where M is infinite. Moreover, we could assume $n = n'$ to simplify the definition.

Theorem 19. *Suppose that L is boundedly discriminable. Then L is independently generated.*

Proof. Without loss of generality we may assume that all the $i \in M$ are minimal in $B(i)$; in particular, the least element of $B(0)$ is $0 \in M$. If $i \notin M$ let $j < i$ be the least element of $B(i)$. Then $j \in M$ and we put $A_j := A_i$.

First we introduce constants for $\{0\} \times A_0$. Next we introduce functions f^k and h^k, $k < n$, as in the previous proof. Finally, for $k < n'$, let z^k be an

$n+1$-ary function, defined similar to g above. Let $P(i,k)$ be the statement: *i is the kth number in $B(i)$*. As before, order the elements of $L_i - A_i$ for each $i \in \omega$, and align the elements of A_i in a sequence k^i_j of length n.

$$z^k((i,j_0),(i,j_1),\cdots,(i,j_n)) := \begin{cases} (i,k^i_0) & \text{if } \langle j_0,j_1,\cdots,j_{n-1}\rangle = \overline{A}_i \\ & \quad P(i,k) \text{ and } j_n = j_{n-1},\ \kappa_i \neq 0 \\ (i,k^i_{p+1}) & \text{if } \langle j_0,j_1,\cdots,j_{n-1}\rangle = \overline{A}_i \\ & \quad P(i,k) \text{ and } j_n = k^i_p,\ p+1 < \kappa_i \\ \text{undefined} & \text{else} \end{cases} \tag{6}$$

It remains that to show that (i) the functions generate L, (ii) L is closed under these functions. (i) is reasonably clear. We generate A_i, $i \in M$, using the functions f^k, and then all the A_i using the g^k as in the previous proof. Finally, the z^k allow to generate all of L_i for $P(i,k)$. Since for each number i there is a $k < n'$ such that $P(i,k)$, we generate L_i from A_i using z^k essentially as we used g. Now on to (ii). Closure under f^k. Consider

$$f^k((i_0,j_0),(i_1,j_1),\cdots,(i_{n-1},j_{n-1}))$$

This is defined only if $i := i_0 = i_1 = \cdots = i_{n-1} \in M$ and $\langle j_0,\cdots,j_{n-1}\rangle \in \overline{A}_i$; and in that case it yields the kth member of $\overline{A}_j$, j the next member of M. Closure under h^k. Pretty much as in the previous proof. Closure under z^k. Consider

$$z^k((i_0,j_0),(i_1,j_1),\cdots,(i_n,j_n))$$

If this is defined, $i := i_0 = i_1 = \cdots = i_n$, i is the kth member of $B(i)$, and $\overline{A}_q = \langle j_0,\cdots,j_{n-1}\rangle$ for some q. If z^k is defined, we know from q alone the identity of i. Thus, L_i is known in the second component. Now if $A_q \subseteq L_i$, then $q \in B(i)$ and so $A_q = A_i$. Now by assumption either $j_n = j_{n-1}$, and we get the least member of $L_i - A_i$ according to the enumeration (if $\kappa_i > 0$). Or else we get the next member according to the enumeration. □

5. Progressive functions

The method has so far been to enumerate L by going through the L_i in increasing order. The interest in this method stems from using grammars to generate languages. We think of a grammar as producing complex expressions from less complex expressions. In that sense, a formation step produces a strictly more complex expression. Consider now an ordering of the E of expressions in increasing complexity. Extend this to a linear order, and number the expressions with natural numbers. We expect now that the meaning of expression j is produced from some of the expressions $0, 1\cdots, j-1$. This way of generating expressions is called *progressive*.

Definition 20. *Let f be a partial n-ary function on ω. A* ***point of progressivity*** *is a vector $\vec{x}$ such that $f(\vec{x}) > \max_{1\leq i\leq n} x_i$ (henceforth simply written $\max \vec{x}$). A* ***point of stagnation*** *is a vector $\vec{x}$ such that $f(\vec{x}) = \max \vec{x}$. A* ***point of regression*** *is a vector $\vec{x}$ such that $f(\vec{x}) < \max \vec{x}$. f is called* ***strictly progressive*** *if it has no points of stagnation or regression. f is* ***weakly progressive*** *if it has no points of regression. Finally, a set of functions is strictly or weakly progressive if all its members are.*

We extend this now to functions on ω^2 as follows. If f is an independent function on ω^2 then it has the form $(f_1(\vec{x}), f_2(\vec{y}))$. We say that f is (strictly, weakly) progressive if f_1 is.

The functions in the previous proofs have generally been weakly progressive. The following theorem shows why we cannot strengthen this to strongly progressive functions.

Theorem 21. *There is a $L \subseteq \omega^2$ which cannot be generated by a finite strictly progressive set of independent partial functions.*

Proof. Suppose that F is a finite set of strictly progressive independent partial functions. Let γ be the cardinality of F, and ζ the maximal arity of these functions. We may wlog assume that $\gamma = \zeta$. Then by progressivity, a member from L_i is obtained by applying a function to the members of $\bigcup_{j<i} L_j$. If their number is bounded by k_i, then there are at most ζk_i^ζ elements. Thus, choose the following sequence of numbers.

$$\begin{aligned} \rho_0 &:= 1 \\ \rho_{i+1} &:= (i+1)\rho_i^{i+1} \end{aligned}$$

This sequence is strictly increasing. Moreover, for each choice of γ and ζ there is i such that

$$\rho_{i+1} > \gamma \left(\sum_{j\leq i} \rho_i \right)^\zeta$$

To see this, note that $\sum_{j\leq i} \rho_i \leq i\rho_i \leq \rho_i^2$, since $\rho_i > i$ (except for $i = 0, 1, 2$). Then

$$\rho_{2\zeta+1} = (2\zeta+1)\rho_{2\zeta}^{2\zeta+1} > (2\zeta+1) \left(\sum_{j\leq i} \rho_{2\zeta} \right)^\zeta$$

Now define

$$L := \{(i, j) : j < \rho_i\}$$

Then $|L_i| = \rho_i$ for all i. It follows that for $i = 2\zeta + 1$ there are not enough functions to generate the elements of $L_{2\zeta+1}$ for the elements with lower index. □

6. Conclusion

We have shown that all countable languages are compositional and autonomous. Moreover, some results have been obtained concerning languages that are independently generated. However, it is open whether all countable languages are independently generated. It is also unclear whether or not allowing partial functions rather than total functions makes a difference.

The conjecture is that there exist nonindependent countable languages. However, no example has been found.

References

Barker, C. and P. Jacobson (eds.). 2007. Direct compositionality. Oxford: Oxford University Press.

Hinzen, W., E. Machery and M. Werning (eds.). 2012. Handbook of compositionality. Oxford: Oxford University Press.

Janssen, T. M. V. 1997. Compositionality. In J. van Benthem and A. ter Meulen (eds.) Handbook of logic and language. Amsterdam & Cambridge, MA: Elsevier & MIT Press. 417–473.

Kracht, M. 2011. Interpreted languages and compositionality. Springer: Dordrecht.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 369–383

Analogy by frequency and functional load: possible reasons for the vowel length neutralisation process in Hungarian?

KATALIN MÁDY
Research Institute for Linguistics
Hungarian Academy of Sciences
mady.katalin@nytud.mta.hu

UWE D. REICHEL
Research Institute for Linguistics
Hungarian Academy of Sciences
uwe.reichel@nytud.mta.hu

KEYWORDS

vowel quantity
neutralisation
type and token frequency
functional load
Hungarian

ABSTRACT

Vowel quantity is phonologically distinctive in Hungarian. Over years, the quantity opposition has become rather unstable in high vowels, especially in unstressed positions. This paper investigates the role of analogy by frequency and that of functional load. Due to the overall higher frequency of short vowels in all vowel classes including low vowels, results provide no evidence for the impact of frequency-based neutralisation. Differences show high functional load for low vowels, but low functional load for high and mid vowels. The potential communicative loss connected to the opposition in low vowels could explain the stability of their opposition.

1. The Hungarian vowel system

Hungarian has traditionally been described as a language with a phonological quantity distinction for both vowels and consonants. While the relevance of this feature for consonants has been questioned by Siptár (1995) based on the small number of minimal pairs and their restricted distribution, the quantity distinction for vowels is unquestionably a relevant feature according to both phonetic and phonological descriptions. The vowels *i–í, ü–ű, u–ú, o–ó, ö–ő*, where the accent sign marks long quantity, not stress, correspond to the vowel qualities /i–iː/, /y–yː/, /u–uː/, /o–oː/, /ø–øː/ and are accounted for as five pairs distinguished primarily by length.

The remaining vowels, *a–á* and *e–é*, are classified in a different manner in form-oriented (phonetic) and function-oriented (phonological) frameworks. Phonetic descriptions take into account that these vowels do not only differ by quantity, but also by quality, the latter being the primary cue. According to traditional descriptions, short *a* is a back, mid-low, rounded vowel /ɔ/, while long *á* corresponds to a central, low, and unrounded /aː/. In Mády (2008) it was argued that vowel height is not necessarily distinctive between *a* and *á*, since the somewhat smaller jaw opening of *a* can be explained by the fact that the vowel is rounded: Hungarian /y/ was also produced with a smaller jaw opening than its unrounded equivalent /i/ in this articulography study. Nevertheless, the *a* vowels /ɒ/ and /aː/ are obviously distinguished by at least one quality feature next to quantity. The same is true for *e* and *é*, of which the short vowel corresponds to a somewhat lowered open-mid front unrounded /ɛ/, and the long one to close-mid front unrounded /eː/. The vowel system of Standard Hungarian does not contain any other vowels such as reduced ones or diphthongs.

Phonologists, on the other hand, encounter phonological processes in which /ɒ/ alternates with /aː/ and /ɛ/ with /eː/ in exactly the same way as short and long mid and high vowels do. One such rule is *Final Stem Vowel Shortening*: in certain stems, a long vowel is replaced by its short counterpart if a given suffix is added to the stem, e.g., ***kút*** /kuːt/ 'well'–*k**u**tak* /kutɒk/ 'wells', ***kéz*** /keːz/ 'hand'–*k**e**zel* /kɛzɛl/ 'handle', ***sár*** /ʃaːr/ 'mud'–*s**a**rat* /sɒrɒt/ 'mud-ACC'. Another example is *Internal Stem Vowel Shortening* that is triggered by certain suffixes such as *-ál*, *-ikus* etc., e.g., *akt**í**v* /ɒktiːv/ 'active'–*akt**i**vál* /ɒktivaːl/ 'activate', *kult**ú**ra* /kultuːrɒ/ 'culture'–*kult**u**rális* /kulturaːliʃ/ 'cultural'. (See Siptár & Törkenczy 2000, 58–62 for a detailed description.) In this paper we will accept the assumption that the vowels /ɒ/–/aː/ and /ɛ/–/eː/ are vowel pairs with quantity as a distinctive feature, regardless of their different qualities, for reasons to be explained below.

Vowel length in Hungarian is encoded by orthography: long vowels are marked by an accent aigu sign, short vowels by the absence of it (e.g., *ó-o*, *ő-ö* for /o/ and /ø/, the umlaut marking front rounded vowels). This has two consequences: first, native Hungarian speakers are consciously aware of vowel quantity, second, the pronunciation norm is conserved by orthography to some extent.

Given this, it might appear surprising that quantity is not consistently realised in the way orthography would suggest. For example, the compound word *k**ó**rház* 'hospital' is often produced as /k**o**rhaːz/, i.e., with a short /o/, in Educated Colloquial Hungarian (a variety widely accepted all over the country), although the lexemes *kór* 'disease' and *ház* 'house' are both pronounced with long vowels. This discrepancy between orthography and the usual pronunciation of a word does not involve mid vowels very often, but it is frequent for the high vowels *u*, *y*, and *i*. While there are some examples where an orthographically short vowel is produced as a long one in colloquial speech (such as *dicsér* 'praise' that is pronounced as /diːʧeːr/ by many speakers), the vast majority of the discrepancies involves cases in which long graphemes are produced as short vowels (e.g., *címke* /ʦimkɛ/ 'label'). The shortening tendency is even more advanced in unstressed syllables. Siptár and Törkenczy (2000) claim that the quantity distinction for high vowels is missing completely in word-final position, at least in Educated Colloquial Hungarian.

In this paper, the relevance of the distribution of short and long vowels is investigated, with a special focus on syllables carrying higher or lower prominence. Hungarian has fixed word-level stress that is always word-initial. Syllables with lexical stress are potential carriers of sentence-level pitch accents. Thus, the distribution of short and long vowels in these syllables might have a different impact on the preservation of the vowel quantity distinction.

The structure of this paper is as follows: in section 2, a short diachronic overview of quantity variation in dialects is given. In section 3, the frequency of short and long vowels in a type and a token word list is analysed. In section 4, the potential interplay between the functional load of a quantity opposition and its preservation is investigated.

2. Dialectal variation

There is considerable variation in the distribution of high short and long vowels across the regional varieties of Hungarian. In large parts of Western Hungary, the vowel system does not contain long high vowels at all (Kálmán 1989). On the other hand, a prevalence of long high vowels can be observed in the Eastern Hungarian dialects, in which many short vowels of the standard variety are lengthened, especially in stressed (i.e., word-initial) syllables. According to Benkő (1957), short vowel lengthening in the Eastern dialects took place from the 16th century on, and the opposite tendency to shorten long vowels in the Western regions is at least this old. Benkő explains the instability of these vowels by the process of the Final Stem Vowel Shortening (see above): while in the Eastern dialects the shortening rule often failed to apply to suffixed stems and resulted in a higher number of long vowels, the Western region shortened vowels in unsuffixed stems analogously to their suffixed forms as opposed to Central Hungarian dialects on which today's standard is based. (E.g., Eastern ***út*** 'road', ***ú**tazik* 'travel', as opposed to Central ***ú**t*, ***u**tazik*, and Western *v**i**z* 'water', *v**i**zes* 'wet' as opposed to Central *v**í**z*, *v**i**zes*.)

Another change in vowel quantity was the shortening of word-final long /uː/ and /yː/ that included large regions both in West and East (in the West it also applied to /iː/). Since word-final vowels in polysyllabic words are always unstressed in Hungarian, Benkő (1957) suggests that the shortening process is due to the missing prominence in these syllables. Although the same process did not take place in Central Hungary, it is remarkable that vowel shortening in unstressed syllables has become part of today's Colloquial Educated Hungarian that is also spoken in the capital Budapest in Central Hungary.

According to Benkő, the quantity change in stressed syllables was triggered by the coexistence of stems with long and short word-final vowels in their unsuffixed and suffixed forms. However, as described above, this rule is not restricted to high vowels, on the contrary, it mostly applies to stems with low vowels (for an exhaustive list, see Siptár 2003, 311). The few stems that include word-final mid vowels are special: in course of the shortening process, a /v/ is inserted after the vowel, e.g., *l**ó*** 'horse' vs. *l**o**vak* 'horses'. Thus, the systematic shortening of stem vowels alone cannot account for the variable quantity of high vowels.

3. Distribution of short and long vowels

3.1. Frequency in types

Another potential reason for the shortening of long high vowels is their lower frequency in the language. According to Gósy (2004), long vowels are far less frequent in Hungarian than short ones. The proportion of long vowels is given with 21% among all vowels, long /yː/ being the most unfrequent one (*ibid.*, 85ff).

A further question is whether the distribution of long and short vowels is identical across word-initial (= stressed and potentially accented) and non-initial (always non-prominent) syllables. A lower occurrence of long vowels in non-initial syllables could explain the shortening tendency in an analogy-based framework.

Type analysis was performed on a word list consisting of 29,245 lemmas based on the lexical entries in *Magyar értelmező kéziszótár* (Pusztai 2003). The list contained word stems (such as *asztal* ‘table’) and derived forms (such as *asztalos* ‘carpenter’), but not words with inflectional suffixes such as plural forms, since the latter are not separate lexicon entries.

Frequency counts for vowels are given in Table 1. Since there is no general agreement on the corresponding IPA symbols in the literature, orthographic symbols are used.

Table 1: Frequency of vowels in lexicon entries in *Magyar értelmező Kéziszótár*

Vowel	Word-initial syllable	Non-initial syllable	Sum
a	5804	10130	15934
á	2121	6387	8508
e	6016	9990	16006
é	1823	3554	5377
i	3926	8420	12346
í	532	1188	1720
o	3673	6766	10439
ó	686	2938	3624
ö	1295	1143	2438
ő	339	1382	1721
u	1749	2742	4491
ú	417	484	901
ü	610	512	1122
ű	221	350	571

Figures 1 and 2 (overleaf) show the frequency of short and long vowels in the dataset, consisting of 85 198 vowels altogether, 29 212 of which occurred in the first syllable of the word (34%). The amount of short vowels was 62 776 (74%).

Given that the proportion of non-initial syllable positions was substantially higher, the ratios reported in the figures were calculated based on the overall count of vowels in word-initial vs. non-initial syllables, e.g., the absolute frequency of short /o/ in the word-initial syllable was divided by the number of all vowels in the same position.

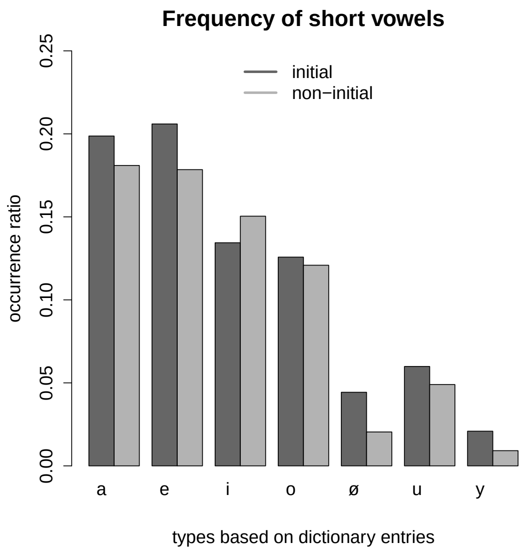


Figure 1: Frequency of short vowels in lemmata. Dark grey: vowels in word-initial syllables, light grey: vowels in non-initial syllables.

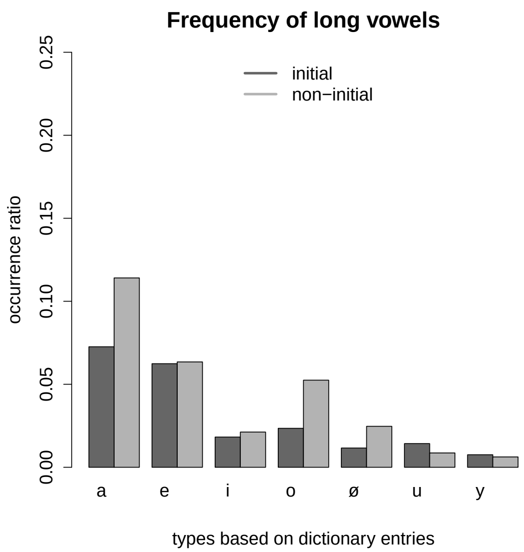


Figure 2: Frequency of long vowels in lemmata. Dark grey: vowels in word-initial syllables, light grey: vowels in non-initial syllables.

The amount of short vowels in word-initial syllable position was higher than in non-initial syllables, except for /i/. At the same time, long vowels occurred more often in non-initial syllables in most vowel pairs. A possible reason for this could be that certain derivational suffixes with high frequency contain long vowels, such as *-ás/-és*, *-ság/-ség*.

Interestingly, vowel frequency did pattern with the three categories high, mid and low. Short /ɛ/ was the most frequent vowel closely followed by short /ɒ/, and also their long equivalents were more frequent than the other five vowels. The low frequency of the other high vowels /u/, /y/ is in line with the analogy hypothesis. On the other hand, the frequency of long /iː/ was almost identical with that of long /øː/, while /i/ alone is involved in the shortening process described in the literature. Thus, the tendency observed in the type-based word list does not support the analogy by frequency hypothesis.

3.2. Frequency in tokens

Type frequency by itself is not necessarily informative about the actual occurrence of long vowels in spoken language for various reasons. First, token frequency is not taken into account. Second, inflexional suffixes that do not occur in the list of lexical entries discussed above contain more often short than long vowels in Hungarian. Third, the frequency of certain types in colloquial speech can substantially differ from the word list discussed in the previous section.

For this reason, spoken language data from a maptask corpus was used to analyse the distribution of short and long vowels in spontaneous speech. The corpus contains data from 27 speakers between 18 and 63 years, including 13 female and 14 male speakers. Dialectal background and social status of speakers differed (see Mády 2010a for more detail).

The database was created for the acoustic analysis of short and long vowels in identical consonantal environments. On one of the maps, a path along various objects was marked. The speaker with this map was supposed to guide the second speaker along this path by verbally explaining the route. As can bee seen in Figure 3, the two maps were not completely identical, resulting in vivid discussions during the recording session. The maps used for the task are shown in Figure 3 (overleaf).

The overall length of the speech material was 115 minutes. Recordings were transcribed into their orthographic form, i.e., according to the grapheme system of Hungarian using the canonical form of words. The material was segmented into word forms. Unfinished forms due to interruption

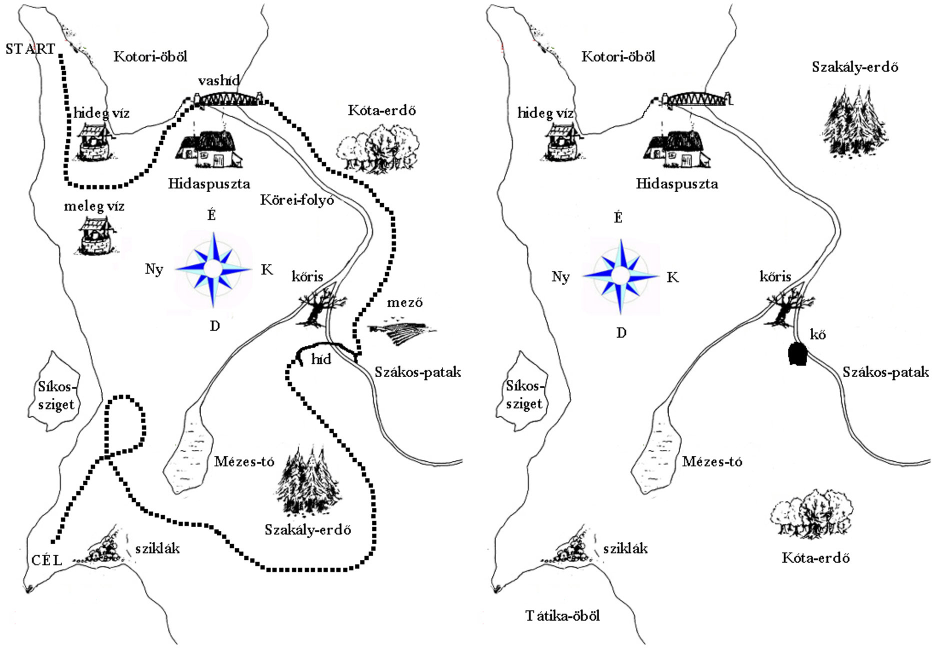


Figure 3: Maps used for the task. Left: first speaker's map, right: second speaker's map.

were removed from the list. Since target words were partly invented geographical names such as *Szákos-patak* referring to a stream that are not part of everyday language usage, proper names were not taken into account for further analysis.

One of our research questions is whether the frequency of short and long vowels located in potentially prominent (i.e., pitch-accented) syllables is identical. However, not every word can carry a pitch accent. Therefore, definite and indefinite articles such as *a* 'the', non-accentable conjunctions such as *és* 'and', *ha* 'if' and modal particles such as *hát* 'well' were excluded from the analysis. Admittedly, this procedure is blind for the presence of pitch accents on these words. For example, the indefinite article *egy* 'a' is homophonous with the numeral *egy* 'one', and the latter usually carries a pitch accent. Since manual checking of the accent patterns was not possible in this case, all tokens of this type were disregarded. Other words that can function both as a content or a function word, e.g., *fog* 'grab' or a future auxiliary, were regarded as potential prominence carrier units and were included in the analysis.

The final dataset contained 17,916 vowels, 9024 of which were located in word-initial syllables (50%), and 13,475 were short ones (75%). The high proportion of vowels in word-initial syllables was due to the overall high occurrence of monosyllabic words such as verbal prefixes.

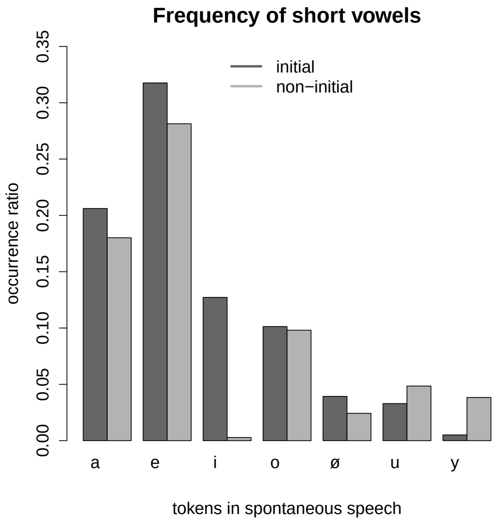


Figure 4: Frequency of short vowels in spontaneous speech. Dark grey: vowels in word-initial syllables, light grey: vowels in non-initial syllables.

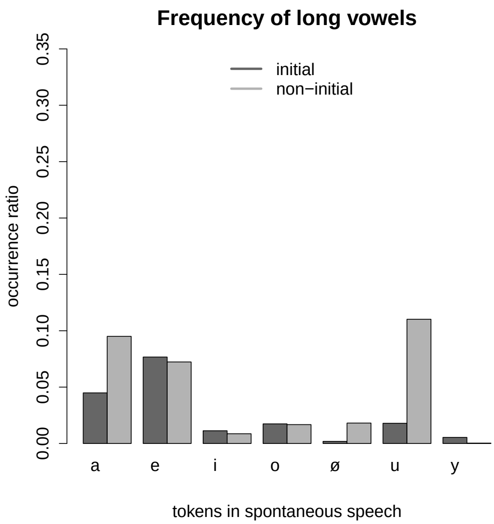


Figure 5: Frequency of long vowels in spontaneous speech. Dark grey: vowels in word-initial syllables, light grey: vowels in non-initial syllables.

The distribution of short vowels was partly different from the frequency counts in the word list. Here, /ɛ/ was by large the most frequent vowel. This is in line with the wide-spread assumption that this sound is the most frequent one in Hungarian, but it differs from frequency data based on the lexicon entries where /ɒ/ was only slightly less frequent than /ɛ/. It is interesting that /i/ was extremely unfrequent in unstressed syllables.

The relative frequencies of long vowels show a similar pattern to type frequencies, with the exception of /uː/.

The distributional data do not favour the hypothesis that the small number of long high vowels could be responsible for the preference for short high vowels. First, /iː/ is not less frequent than /øː/. Second, not only high, but all long vowels are less frequent both in stressed and unstressed positions than short ones.

4. Functional load of quantity oppositions

Next to the vowel frequency analyses we investigated whether the functional load of a quantity opposition could account for its preservation. We hypothesise that oppositions with a high functional load are more stable than oppositions for which the functional load is low. The importance of the opposition is quantified by two measures described in the following.

4.1. Functional load

The functional load (FL) of a phonological opposition of the phonemes a and b is related to the number of contrasts this opposition is responsible for in a language L. The information-theoretic definition adopted here was first introduced in Hockett (1967):

$$\mathrm{FL}(a,b) \quad = \quad \frac{H(L) - H(L_{a=b})}{H(L)}$$

$H(L)$ is the entropy of a language L. $L_{a=b}$ denotes a language lacking an opposition of a and b. $\mathrm{FL}(a,b)$ thus stands for the relative amount of information loss resulting from such a merging, reflecting the increase of homophones.

L and $L_{a=b}$ are the sets of word types w of *Magyar értelmező kéziszótár* before and after vowel merging, respectively. Merging or neutralisation means that in the second lexicon, long vowels were replaced by their short counterpart in all stems. From the word type frequencies contained in this dictionary, maximum likelihood probabilities were calculated in order to derive the entropies for L and $L_{a=b}$ as follows:

$$H(L) = -\sum_{w \in L} p(w) \log_2 p(w)$$
$$H(L_{a=b}) = \sum_{w \in L_{a=b}} p(w) \log_2 p(w)$$

The frequencies for merged types in $L_{a=b}$ were simply obtained by summing up the frequencies of all types of L undergoing this merging after the neutralisation of the opposition of a and b.

4.2. Type number ratio

Since the FL is calculated over the entire lexicon, it does not normalise for vowel-related frequencies, that in turn determine the number of resulting homophones after quantity merging. FL is positively correlated with vowel frequency, since the merging of frequent vowels results in more homophones so that their quantity opposition receives a high functional load.

In order to reduce this frequency bias, we additionally calculated the ratio of those types only, that are affected by a vowel quantity merging. As an example, for the merging of /uː/ we considered only those words that contain the letter *u* and/or *ú*. For these types we derived the ratio N_a/N_b, where N_b denotes the number of types before merging and N_a denotes the number of types after merging.

4.3. Word stress

In order to test the impact of word stress on quantity merging, we applied the two measures for three different vowel merging scenarios: (1) in all syllables, (2) in the word-stressed (initial) syllable only, and (3) in all non-stressed (non-initial) syllables only.

4.4. Results

The functional loads and type number ratios for each vowel pairing are shown in Figures 6 and 7, respectively. Important oppositions are indicated by a high functional load and a low type number ratio.

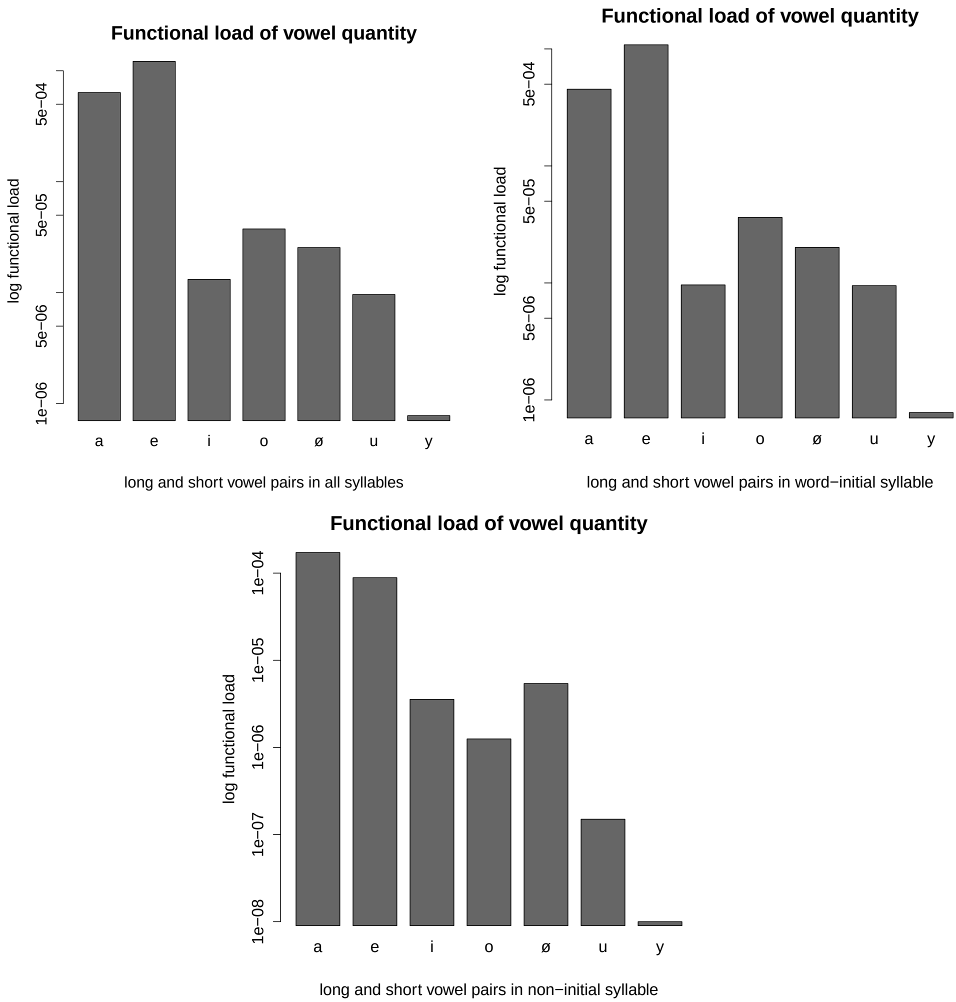


Figure 6: Functional loads of vowel quantity oppositions over the entire word (top-left), in word-initial stressed position (top-right), and in non-initial unstressed position (bottom)

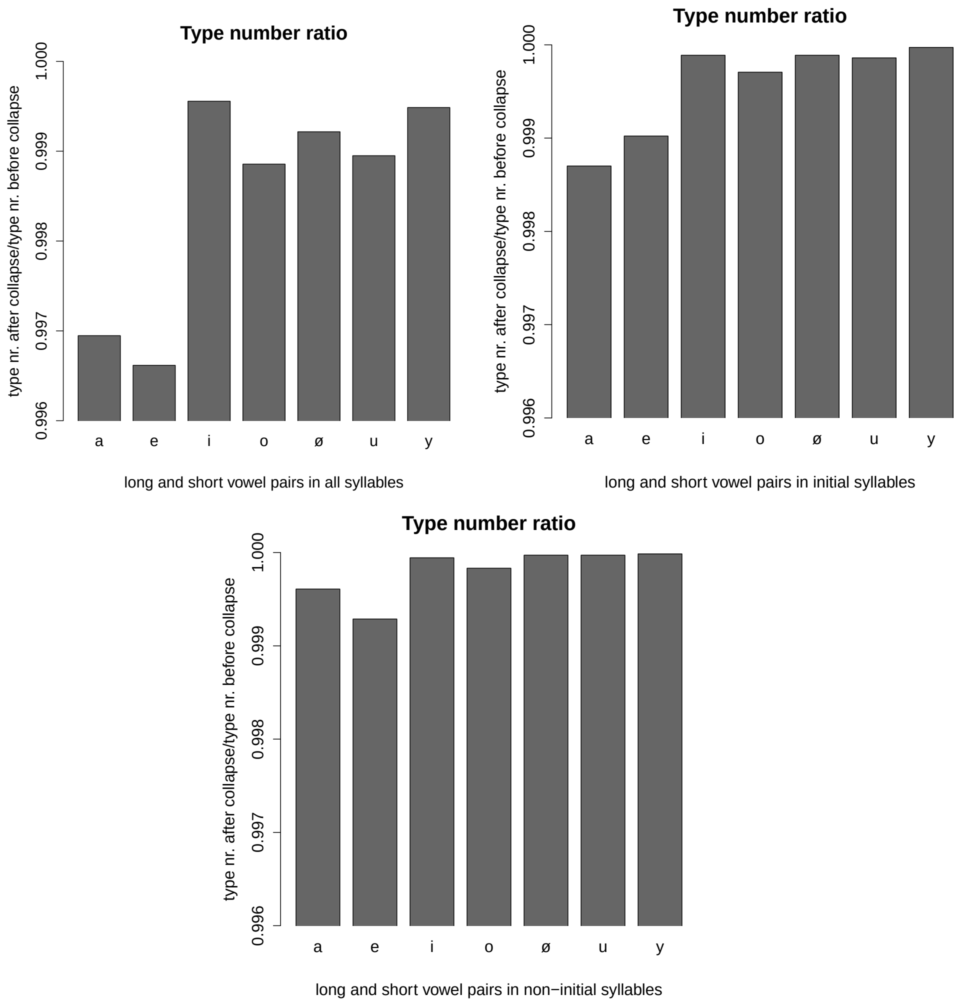


Figure 7: Type number ratios of vowel quantity oppositions over the entire word (top-left), in word-initial stressed position (top-right), and in non-initial unstressed position (bottom)

Based on Figure 6 for the impact of functional load on quantity preservation, the following conclusions can be drawn:

- The quantity oppositions for /e, a/ have the highest functional loads, which – in line with our hypothesis – prevents them to undergo quantity merging. This merging would result in a significant decrease in lexical contrasts and therefore an increase in ambiguity.
- Over the entire word, /i, u, y/ quantity oppositions have the lowest functional loads. Thus it is not crucial to maintain these oppositions, and indeed, these vowels are least stable in preserving them.
- Also word-initially the functional loads of /i, u, y/ quantity oppositions are the lowest, so that the absence of these oppositions e.g., in Western Hungarian does not lead to a communicative loss.
- However, in non-initial syllables also the quantity opposition for /ø/ has a low functional load, but against the expectation for this vowel the quantity opposition is maintained.

For the type number ratios shown in Figure 7 we obtained the same tendencies.

- For /a, e/ the lowest ratios were measured, again well explaining the stability of their quantity contrasts in order not to drastically increase the number of homophones.
- /i, u, y/ show high ratios, especially in non-initial syllables, indicating only a negligible increase of ambiguity in case of quantity merging.
- However, high ratios are given also for /o, ø/.

5. Discussion and conclusions

Vowel statistics, i.e., analogy by frequency did not turn out to play a crucial role in explaining varying degrees of the stability of quantity oppositions. The functional load of an opposition, however, was found to have an impact on maintaining quantity contrasts. A high functional load is a sufficient motivation to maintain such contrasts. The reverse case, i.e., when a low functional load leads to quantity merging, holds for high vowels.

The quantity opposition in mid vowels might be subject to an ongoing sound change process. Perception experiments in Mády (2010b) and Mády (2012) show that the perceptual boundary between long and short /o/ in word-final position is shifted towards the short vowel in young speakers, but not in the older group. Young participants categorised both shorter and more centralised /o/ segments as long vowels, whereas a segment had to be longer and more centralised to be identified as a long /oː/ by listeners above 50 years. Thus, the quantity distinction might become less stable also for mid vowels within a certain time range. This development would again be well explainable by the low functional load of mid-vowel quantity oppositions.

Acknowledgements

Data for the maptask corpus were recorded at the Laboratory of Speech Acoustics at the Technical University of Budapest. We highly appreciate the help of Klára Vicsi, György Szaszák and their colleagues.

References

Benkő, L. 1957. Magyar nyelvjárástörténet [Hungarian historical dialectology]. Budapest: Tankönyvkiadó.

Gósy, M. 2004. Fonetika, a beszéd tudománya [Phonetics, the science of speech]. Budapest: Osiris Kiadó.

Hockett, Ch. F. 1967. The quantification of functional load. Word 23. 320–339.

Kálmán, B. 1989. Nyelvjárásaink [Hungarian dialects]. Fifth edition. Budapest: Tankönyvkiadó. [Our dialects].

Mády, K. 2008. Magyar magánhangzók vizsgálata elektromágneses artikulográffal gyors és lassú beszédben [Investigation of Hungarian vowels in normal and fast speech: An articulography study]. Beszédkutatás 2008. 52–66.

Mády, K. 2010a. Hungarian vowel quantity neutralisation as a potential social marker. Acta Linguistica Hungarica 57. 167–188.

Mády, K. 2010b. Shortening of long high vowels in Hungarian: A perceptual loss? In C. C. Silvia Calamai and L. Ciucci (eds.) Sociophonetics, at the crossroads of speech variation, processing and communication Papers from the international workshop "Sociophonetics, at the crossroads of speech variation, processing and communication". Pisa: Scuola Normale Superiore. 41–44.

Mády, K. 2012. Implicit and explicit language attitude in a sound change process. In Proceedings of the 2nd Workshop on Sound Change, 2–4 May 2012, Kloster Seeon.

Pusztai, F. (ed.). 2003. Magyar értelmező kéziszótár. 2. átdolgozott kiadás [A concise dictionary of Hungarian. 2nd, revised edition]. Budapest: Akadémiai Kiadó.

Siptár, P. 1995. A magyar mássalhangzók fonológiája [The phonology of Hungarian consonants] (Linguistica. Studia et dissertationes 18). Budapest: Hungarian Academy of Sciences, Research Institute for Linguistics.

Siptár, P. 2003. Hangtan [Phonology]. In K. É. Kiss, F. Kiefer and P. Siptár: Új magyar nyelvtan [A new Hungarian grammar]. Third edition. Budapest: Osiris Kiadó. 293–390.

Siptár, P. and M. Törkenczy. 2000. The phonology of Hungarian. Oxford: Oxford University Press.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 385–398

Do multi-sense embeddings learn more senses?

An evaluation in linear translation

MÁRTON MAKRAI
Research Institute for Linguistics,
Hungarian Academy of Sciences
makrai.hlt@gmail.com

VERONIKA LIPP
Research Institute for Linguistics,
Hungarian Academy of Sciences
lipp.veronika@nytud.mta.hu

KEYWORDS

word embedding
ambiguity
translation
nearest neighbors
Dirichlet Process

ABSTRACT

We analyze whether different sense vectors of the same word form in multi-sense word embeddings correspond to different concepts. On the more technical side of embedding-based dictionary induction, we also test whether the orthogonality constraint and related vector preprocessing techniques help in reverse nearest neighbor search. Both questions receive a negative answer.

Word sense induction (WSI) is the task of discovering senses of words without supervision (Schütze 1998). Recent approaches include multi-sense word embeddings (MSEs), i.e., vector space models of word distribution with more vectors for ambiguous words. In MSEs, each vector is supposed to correspond to a different word sense, but in practice models frequently have different sense vectors for the same word form without an interpretable difference in meaning.

In Borbély et al. (2016), we proposed a cross-lingual method for the evaluation of sense resolution in MSEs. The method is based on the principle that words may be ambiguous to the extent to which their postulated senses translate to different words in some other language. For the translation of words, we applied the method by Mikolov et al. (2013b) who train a translation mapping from the source language embedding to the target as a least-squares regression supervised by a seed dictionary of the few thousand most frequent words. The translation of a source word vector is the nearest neighbor of its image by the mapping in the target space. In the multi-sense setting, we have translated from MSEs. (The target embedding remained single-sense.)

Section 1 discusses our linguistic motivation and section 2 introduces MSEs. In section 3, we elaborate on the cross-lingual evaluation. Part of the evaluation task is to decide on empirical grounds whether different good translations of a word are synonyms or translations in different senses. Reverse nearest neighbor search, the orthogonality constraint on the translation mapping, and related techniques are also discussed. Section 4 offers experimental results with quantitative and qualitative analysis. It should be noted that our evaluation is not very strict, but rather a process of looking for something conceptually meaningful in present-day unsupervised MSE models. We make our Hungarian multi-sense embeddings[1] and the code for these experiments[2] available on the web.

1. Towards a less *delicious* inventory

We emphasize that our evaluation proposal probes an aspect of MSEs, *semantic resolution*, which is not well measured by the well-known word sense disambiguation (WSD) task that aims at classifying occurrences of a word form to different elements of a sense inventory pre-defined by some experts. Our goal in WSI is to probe the granularity of the inventory itself. The differentiation of word senses, as already noted in Borbély et al. (2016), is fraught with difficulties, especially when we wish to distinguish homophony, i.e., using the same written or spoken form to express different concepts, such as Russian *mir* 'world' and *mir* 'peace' from polysemy, where speakers feel that the two senses are very strongly connected, such as in Hungarian *nap* 'day' and *nap* 'sun'.

The goal of WSI can be set at two levels. We may more modestly aim to distinguish homophony from polysemy. Ideally, we could even differentiate between metonymy and metaphor, two subtypes of polysemy, discussed in more detail in the next section.

[1] https://hlt.bme.hu/en/publ/makrai17

[2] https://github.com/makrai/wsi-fest

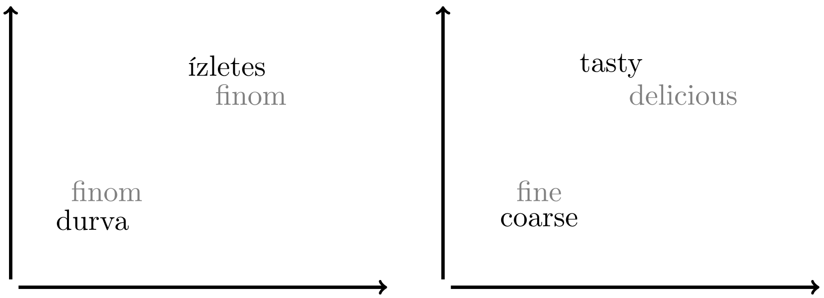


Figure 1: Linear translation of word senses. The Hungarian word *finom* is ambiguous between 'fine' and 'delicious'.

1.1. Lexicographic background

Lexical ambiguity is linguistically subdivided into two main categories: *homonymy* and *polysemy* (Cruse 2004). Homonymous words have semantically unrelated and mutually incompatible meanings, such as $punch_1$, which means 'a blow with a fist', and $punch_2$, which means 'a drink'. Some have described such homonymous word meanings as essentially distinct words that accidentally have the same phonology (Murphy 2002). Polysemous words, on the other hand, have semantically related or overlapping senses (Cruse 2004; Jackendoff 2002; Pustejovsky 1995), such as *mouth* meaning both 'organ of body' and 'entrance of cave'.

Two criteria have been proposed for the distinction between homonymy and polysemy. The first criterion has to do with the *etymological* derivation of words. Words that are historically derived from distinct lexical items are taken to be homonymous. However, the etymological criterion is not always decisive. One reason is that there are many words whose historical derivation is uncertain. Another reason is that it is not always very clear how far back we should go in tracing the history of words (Lyons 1977).

The second criterion for the distinction between homonymy and polysemy has to do with the *relatedness/unrelatedness of meaning*. The distinction between homonymy and polysemy seems to correlate with the native speaker's feeling that certain meanings are connected and that others are not. Generally, unrelatedness in meaning points to homonymy, whereas relatedness in meaning points to polysemy. However, in a large number of cases, there does not seem to be an agreement among native speakers as to whether the meanings of the words are related. So, it seems that there is not a clear dichotomy between homonymy and polysemy, but rather a continuum from "pure" homonymy to "pure" polysemy (Lyons 1977).

Most discussions about lexical ambiguity, within theoretical and computational linguistics, concentrate on polysemy, which can be further divided into two types (Apresjan 1974; Pustejovsky 1995). The first type of polysemy is motivated by *metaphor (irregular polysemy)*. In metaphorical polysemy, a relation of analogy is assumed to hold between the senses of the word. The basic sense of metaphorical polysemy is literal, whereas its secondary sense is figurative. For example, the ambiguous word *eye* has the literal basic sense 'organ of the body' and the figurative secondary sense 'hole in a needle.' The other type of polysemy is motivated by *metonymy (regular polysemy)*. In metonymy, the relation that is assumed to hold between the senses of the word is that of contiguity or connectedness. In metonymic polysemy, both the basic and the secondary senses are literal. For example, the ambiguous word *chicken* has the literal basic sense referring to the animal and the literal secondary sense of the meat of that animal.

2. Multi-sense word embeddings

Vector-space language models with more vectors for each meaning of a word originate from Reisinger & Mooney (2010). Huang et al. (2012) trained the first neural-network-based MSE. Both works use a uniform number of clusters for all words that they select before training as potentially ambiguous. The first system with adaptive sense numbers and an effective open-source implementation is a modification of skip-gram (Mikolov et al. 2013c), *multi-sense* skip-gram by Neelakantan et al. (2014), where new senses are introduced during training by thresholding the similarity of the present context to earlier contexts.

Bartunov et al. (2016) and Li & Jurafsky (2015) improve upon the heuristic thresholding by formulating text generation as a Dirichlet process. In `AdaGram` (Bartunov et al. 2016), senses may be merged as well as allocated during training. `mutli`-sense skip-gram[3] (Li & Jurafsky 2015) applies the Chinese restaurant process formalization of the Dirichlet process. `neela`, `AdaGram`, and `mutli` have a parameter for semantics resolution (more or less senses): λ, α, and γ, respectively.

[3] Note the $l \leftrightarrow t$ metathesis in the name of the repo which is the only way of distinguishing it from the other two multi-sense skip-gram models.

MSEs are still in the research phase: Li & Jurafsky (2015) demonstrate that, when meta-parameters are carefully controlled for, MSEs introduce a slight performance boost in semantics-related tasks (semantic similarity for words and sentences, semantic relation identification, part-of-speech tagging), but similar improvements can also be achieved by simply increasing the dimension of a single-sense embedding.

3. Linear translation from MSEs

Mikolov et al. (2013b) discovered that embeddings of different languages are so similar that a linear transformation can map vectors of the source language words to the vectors of their translations.

The method uses a seed dictionary of a few thousand words to learn translation as a linear mapping $W : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ from the source (monolingual) embedding to the target: the translation $z_i \in \mathbb{R}^{d_2}$ of a source word $x_i \in \mathbb{R}^{d_1}$ is approximately its image Wx_i by the mapping. The translation model is trained with linear regression on the seed dictionary

$$\min_W \sum_i ||Wx_i - z_i||^2$$

and can be used to collect translations for the whole vocabulary by choosing z_i to be the nearest neighbor (NN) of Wx_i. We follow Mikolov et al. (2013b) in (i) using different metrics, Euclidean distance in training and cosine similarity in collection of translations, and in (ii) training the source model with approximately three times greater dimension than that of the target embedding.

In a multi-sense embedding scenario, Borbély et al. (2016) take an MSE as the source model, and a single-sense embedding as target. The quality of the translation has been measured by training on the most frequent 5k word pairs and evaluating on another 1k seed pairs.

3.1. Reverse nearest neighbor search

A common problem when looking for nearest neighbors in high-dimensional spaces (Radovanović et al. 2010; Suzuki et al. 2013; Tomašev & Mladenic 2013), and especially in embedding-based dictionary induction (Dinu et al. 2015; Lazaridou et al. 2015) is when there are *hubs*, data points (target words) returned as the NN (translation) of many points (Wxs), resulting in incorrect hits (translations) in most of the cases. Dinu et al. (2015) attack the problem with a method they call *global correction*. Here, instead of the original NN, which we will call *forward* NN search to contrast with the more sophisticated method, they first rank source words by their similarity to target words. In *reverse* nearest neighbor (rNN) search, source words are translated to the target words to which they have the lowest (forward) NN rank.[4]

In reverse NN search, we restricted the vocabulary to the some tens of thousands of the most frequent words. We introduced this restriction for memory saving, because the $|V_{sr}| \times |V_{tg}|$ similarity matrix has to be sorted column-wise for forward and row-wise for reverse ranking, so at some point of the computation we keep the whole integer matrix of forward NN ranks in memory. It turned out that the restriction makes the results better: a vocabulary cutoff of $2^{15} = 32768$ both on the source and the target size yields slightly better results (74.3%) than the more ambitious $2^{16} = 65536$ (73.9%). This is not the case for forward NN search, where accuracy increases with vocabulary limit (but remains far below that of reverse NN).

3.2. Orthogonal restriction and other tricks

Xing et al. (2015) note that the original linear translation method is theoretically inconsistent due to its being based on three different similarity measures: `word2vec` itself uses the dot-product of unnormalized vectors, the translation is trained based on Euclidean distance, and neighbors are queried based on cosine similarity. They make the framework more coherent by length-normalizing the embeddings, and restricting W to preserve vector length: their matrix W is orthogonal, i.e., the mapping is a rotation. Faruqui & Dyer (2014) achieve even better results by mapping the two embeddings to a lower-dimensional bilingual space with canonical correlation analysis. Artetxe et al. (2016) analyze elements of these two works both theoretically and empirically, and find a combination that improves upon dictionary generation and also preserves analogies Mikolov (2013d) like

$$\mathbf{woman} + \mathbf{king} - \mathbf{man} \approx \mathbf{queen}$$

among the mapped points Wx_i. They find that the orthogonality constraint is key to preserve performance in analogies, and it also improves bilingual performance. In their experiments, length normalization, when followed by centering the embeddings to $\mathbf{0}$ mean, obtains further improvements in bilingual performance without hurting monolingual performance.

[4] If more target words have the same forward rank, Dinu et al. (2015) make the decision based on cosine similarity. This tie breaking has not proven useful in our experiments.

4. Experiments

4.1. Data

We trained `neela`, `AdaGram` and `mutli` models on (original and stemmed[5] forms of) two semi-gigaword (.7–.8 B words) Hungarian corpora, the Hungarian Webcorpus (Webkorpusz, Halácsy et al. 2004) and (the non-social-media part of) the Hungarian National Corpus (HNC, Oravecz et al. 2014). We used Wiktionary as our seed dictionary, extracted with `wikt2dict`[6] (Ács et al. 2013). We tried several English embeddings as target, including the 300 dimensional skip-gram with negative sampling model `GoogleNews` released with `word2vec` (Mikolov et al. 2013a),[7] and those released with GloVe (Pennington et al. 2014).[8] We report the best results, which were obtained with the release GloVe embeddings trained on 840 B words in 300 dimensions.

4.2. Orthogonal constraint

We implemented the orthogonal restriction by computing the singular value decomposition

$$U\Sigma V = S_t^\top T_t$$

where S_t and T_t are the matrices consisting of the embedding vectors of the training word pairs in the source and the target space respectively, and taking

$$W = U\mathbf{1}V$$

where $\mathbf{1}$ is the rectangular identity matrix of appropriate shape.

Table 1 (overleaf) shows the effect of these factors. Precision in forward NN search follows a similar trend to that in Xing et al. (2015) and Artetxe (2016): the best combination is an orthogonal mapping between length-normalized vectors; however, centering did not help in our experiments. Reverse NNs yield much better results than the simpler method, but none of the orthogonality-related techniques give further improvement here. The cause of reverse NN's apparent insensitivity to length may be the topic of further research.

[5] Follow-up work reported in section 4.5 applied a third option in preprocessing.

[6] https://github.com/juditacs/wikt2dict

[7] https://code.google.com/archive/p/word2vec/

[8] https://nlp.stanford.edu/projects/glove/

		8192				16384				32768			
		general linear		orthogonal		general linear		orthogonal		general linear		orthogonal	
		any	disamb	any	disamb	any	disamb	any	disamb	any	disamb	any	disamb
fwd	vanilla	28.7%	2.40%	32.1%	2.40%	36.2%	3.40%	42.0%	4.70%	36.7%	4.20%	44.5%	6.00%
fwd	normalize	28.2%	2.20%	**33.7%**	3.40%	35.1%	2.80%	**44.4%**	5.80%	36.6%	3.80%	**48.2%**	6.00%
fwd	+ center	26.6%	2.10%	32.8%	2.90%	32.9%	2.70%	42.0%	4.50%	34.6%	3.50%	43.9%	5.50%
rev	vanilla	**53.8%**	11.85%	51.7%	11.37%	**58.3%**	11.99%	56.6%	12.59%	**74.3%**	23.60%	73.6%	22.30%
rev	normalize	53.3%	11.61%	50.0%	10.90%	58.0%	12.35%	56.5%	12.59%	73.7%	24.20%	72.8%	22.10%
rev	+ center	51.7%	11.37%	53.3%	11.14%	57.1%	11.99%	57.7%	12.35%	69.7%	22.20%	73.5%	23.00%

Table 1: Precision@10 of forward and reverse NN translations with and without the orthogonality constraint and related techniques at vocabulary cut-offs 8192 to 32768. `any` and `disamb` are explained in section 4.3. The source has been an `AdaGram` model in 800 dimensions, $\alpha = .1$, trained on Webkorpusz with the vocabulary cut off at 8192 sense vectors.

4.3. Results

We evaluate MSE models in two ways, referred to as `any` and `disamb`. The method `any` has been used for tuning the (meta)parameters of the source embedding and to choose the target: a traditional, single-sense translation has been trained between the first sense vector of each word form and its translations. (If the training word is ambiguous in the seed dictionary, all translations have been included in the training data.) Exploiting the multiple sense vectors, one word can have more than one translation. During the test, a source word was accepted if `any` of its sense vectors had at least one good translation among its k reverse nearest neighbors (rNN@k).

In `disamb`, we used the same translation matrix as in `any`, and inspected the translations of the different sense vectors to see whether the vectors really model different senses rather than synonyms. The lowest requirement for the non-synonymy of sense vectors s_1, s_2 is that the sets of corresponding good rNN@k translations are different. The ratio of words satisfying this requirement among all words with more than one sense vector is shown as `disamb` in Table 2.

The values in Table 2 are low. This can in part be due to that the `neela` and the `mutli` models were trained with lower dimension than the best-performing model, so results here are not comparable among these different architectures. Follow-up experiments (conducted after the paper review) are reported in section 4.5.

	dim	α/γ	p	m	any	disamb
HNC	800	.02		100	48.5%	7.6%
`neela` Wk	300	–	2	big	54.0%	12.4%
HNC stem	800	.05		big	55.1%	10.4%
HNC	160	.05	3	200	62.2%	15.0%
`mutli` Wk	300	.25		71	62.9%	17.4%
Webkorpusz	800	.05		100	65.9%	17.4%
HNC	600	.05	5	100	68.6%	16.6%
HNC	600	.1	3	50	69.1%	18.8%
Webkorpusz	800	.1		100	73.9%	23.9%

Table 2: Our measures, `any` and `disamb`, for different MSEs. The source embedding has been trained with `AdaGram`, except for when indicated otherwise (`neela`, `mutli`). The meta-parameters are *dim*ension, the resolution parameter (α in `AdaGram` and γ in `mutli`), the maximum number of *p*rototypes (sense vectors), and the vocabulary cutoff (*m*in-freq, the two models with *big* have practically no cut-off).

Table 3 (overleaf) shows the successfully disambiguated words sorted by the cosine similarity s of good rNN@1 translations of different sense vectors. (We found that most of the few cases when there are more than two sense vectors with a good rNN@1 translation are due to the fact that the seed dictionary contains some non-basic translation, e.g., *kapcsolat* 'relationship, conjunction' has 'affair' among its seed translations. In these cases, we chose two sense vectors arbitrarily.) Relying on s is similar to the monolingual setting of clustering the sense vectors for each word, but here we restrict our analysis to sense vectors that prove to be sensible in linear translation.

We see that most words with $s < .25$ are really ambiguous from a standard lexicographic point of view, but the translations with $s > .35$ tend to be synonyms instead.

	s			covg
E	−0.04849	függő	addict, aerial	0.4
S	0.01821	alkotó	constituent, creator	0.5
S	0.05096	előzetes	preliminary, trailer	1.0
S	0.0974	kapcsolat	affair, conjunction, linkage	0.33
I	0.1361	kocsi	coach, carriage	1.0
S	0.136	futó	runner, bishop	1.0
S	0.1518	keresés	quest, scan	0.67
S	0.1574	látvány	outlook, scenery, prospect	0.6
S	0.1626	fogad	bet, greet	1.0
S	0.1873	induló	march, candidate	1.0
I	0.187	nemes	noble, peer	0.67
E	0.1934	eltérés	variance, departure	0.4
E	0.1943	alkalmazás	employ, adaptation	0.33
S	0.2016	szünet	interval, cease, recess	0.43
E	0.2032	kezdeményezés	initiation, initiative	1.0
S	0.2052	zavar	disturbance, annoy, disturb, turmoil	0.57
S	0.2054	megelőző	preceding, preventive	0.29
IE	0.2169	csomó	knot^I, lump^I, mat^E	1.0
E*	0.21	remény	outlook, promise, expectancy	0.6
S	0.2206	bemutató	exhibition, presenter	0.67
E	0.2208	egyeztetés	reconciliation, correlation	0.5
S	0.237	előadó	auditorium, lecturer	0.67
E	0.2447	nyilatkozat	profession, declaration	0.4
I	0.2494	gazda	farmer, boss	0.67
I	0.2506	kapu	gate, portal	1.0
I	0.2515	előbbi	anterior, preceding	0.67
I	0.2558	kötelezettség	engagement, obligation	0.67
E	0.265	hangulat	morale, humour	0.5
E	0.2733	követ	succeed, haunt	0.67
SE	0.276	minta	norm^S, formula^E, specimen^S	0.75
S	0.2807	sorozat	suite, serial, succession	1.0
S	0.2935	durva	coarse, gross	0.18
I	0.3038	köt	bind, tie	0.67
E	0.3045	egyezmény	treaty, protocol	0.67
I	0.3097	megkülönböztetés	discrimination, differentiation	0.5
I	0.309	ered	stem, originate	0.5
I	0.319	hirdet	advertise, proclaim	1.0
E	0.3212	tartós	substantial, durable	1.0
I	0.3218	ajánlattevő	bidder, supplier, contractor	0.6
I	0.3299	aláírás	signing, signature	0.67
I	0.333	bír	bear, possess	1.0
I	0.3432	áldozat	sacrifice, victim, casualty	1.0
IE	0.3486	kerület	ward^I, borough^I, perimeter^E	0.3
I	0.3486	utas	fare, passenger	1.0
I	0.3564	szigorú	stern, strict	0.5
I	0.3589	bűnös	sinful, guilty	0.5
I	0.3708	rendes	orderly, ordinary	0.5
I	0.3824	eladó	salesman, vendor	0.5
I	0.3861	enyhe	tender, mild, slight	0.6
I	0.3897	maradék	residue, remainder	0.33
I	0.3986	darab	chunk, fragment	0.4
E	0.4012	hiány	poverty, shortage	0.5
I	0.4093	kutatás	exploration, quest	0.5
⋮				
⋮				
I	0.4138	tanítás	tuition, lesson	0.67
I	0.4196	őszinte	frank, sincere	0.67
I	0.4229	környék	neighborhood, surroundings, vicinity	0.38
I	0.4446	ítélet	judgement, sentence	0.67
I	0.4501	gyerek	childish, kid	0.67
I	0.4521	csatorna	ditch, sewer	0.4
I	0.4547	felügyelet	surveillance, inspection, supervision	0.43
E	0.4551	ritka	rare, odd	0.5
S	0.4563	szerető	fond, lover, affectionate, mistress	0.67
I	0.4608	szeretet	affection, liking	0.67
I	0.4723	vizsgálat	inquiry, examination	0.67
I	0.4853	tömeg	mob, crowd	0.5
I	0.4903	puszta	pure, plain	0.22
I	0.4904	srác	kid, lad	1.0
I	0.4911	büntetés	penalty, sentence	0.29
I	0.4971	képviselő	delegate, representative	0.67
I	0.4975	határ	boundary, border	0.67
I	0.5001	drága	precious, dear, expensive	1.0
S	0.5093	uralkodó	prince, ruler, sovereign	0.5
I	0.5097	válás	separation, divorce	0.67
I	0.5103	ügyvéd	lawyer, advocate	0.67
I	0.5167	előnyös	advantageous, profitable, favourable	1.0
I	0.5169	merev	rigid, strict	1.0
I	0.5204	nyíltan	openly, outright	1.0
I	0.5217	noha	notwithstanding, albeit	1.0
I	0.5311	hulladék	litter, garbage, rubbish	0.43
I	0.5311	szemét	litter, garbage, rubbish	0.43
I	0.5612	kielégítő	satisfying, satisfactory	1.0
E	0.5617	vicc	joke, humour	1.0
I	0.5737	szállító	supplier, vendor	1.0
I	0.5747	óvoda	nursery, daycare, kindergarten	1.0
I	0.5754	hétköznapi	mundane, everyday, ordinary	0.75
I	0.5797	anya	mum, mummy	1.0
I	0.5824	szomszédos	neighbouring, neighbour	0.4
E	0.5931	szabadság	liberty, independence	1.0
I	0.6086	lelkész	pastor, priest	0.4
I	0.6304	fogalom	notion, conception	1.0
I	0.6474	fizetés	salary, wage	0.67
I	0.6551	táj	landscape, scenery	1.0
I	0.6583	okos	clever, smart	0.67
I	0.6707	autópálya	highway, motorway	0.5
I	0.6722	tilos	prohibited, forbidden	1.0
I	0.6811	bevezető	introduction, introductory	1.0
I	0.7025	szövetség	coalition, alliance, union	0.75
I	0.7065	fáradt	exhausted, tired, weary	1.0
I	0.7066	kiállítás	exhibit, exhibition	0.67
I	0.7135	hirdetés	advert, advertisement	1.0
I	0.7147	ésszerű	rational, logical	1.0
I	0.7664	logikai	logic, logical	1.0
I	0.7757	szervez	organise, organize, arrange	1.0
I	0.8122	furcsa	strange, odd	0.4
I	0.8277	azután	afterwards, afterward	0.67
I	0.8689	megbízható	dependable, reliable	0.67

Table 3: Hungarian words with the rNN@1 translations of their sense vectors. The first column is a post-hoc annotation by András Kornai (*E* error in translation, *I* identical, *S* separate meanings), *s* is the cosine similarity of the translations, *covg* denotes the coverage of the @1 translations over all gold (good) translations. * = the basic translation *hope* is missing.

4.4. Part of speech

The clearest case of homonymy is when unrelated senses belong to different parts of speech (POSs), and the translations reflect these POSs, e.g., *nő* 'woman; increase' or *vár* 'wait; castle'.[9] In purely semantic approaches, like `4lang` (Kornai 2018; Kornai et al. 2015), POS-difference alone is not enough for analyzing a word as ambiguous, e.g., we see the only difference between the noun and participle senses of *alkalmazott*, 'employee; applied' as *employ*ment being the *appli*cation of people for work; in the case of *belső* 'internal; interior', the noun refers to the part of a building described by the adjective.

More interesting are word forms with related senses in the same POS, e.g., *cikk*, 'item; article' (an article is an item in a newspaper); *eredmény*, 'score; result' (a score is a result measured by a number); *magas*, 'tall; high' (tall is used for people rather than high); or *idegen*, 'strange, alien; foreign', where the English translations are special cases of 'unfamiliar' (person versus language).

4.5. Follow-up experiments

After the compilation of the Festschrift, we trained models that enable a more fair comparison of `AdaGram` and `mutli` in terms of semantic resolution: we trained 600-dimensional models for Hungarian to have the 2:1 ratio between the source and the target dimension that has been reported to be optimal for this task (Mikolov et al. 2013b; Makrai in preparation). This time we used the de-glutinized version (Borbély et al. 2016; Nemeskey 2017) of the Hungarian National corpus for better morphological generalization.

We can see in Table 4 (overleaf) that there is a trade-off between the two measures, which may be interpreted to indicate that the more specific a vector is, the easier it is to translate, but if the vectors are too specific, then the translations may coincide.[10]

As a direction for future research, the analysis of the observed and inferred number of word senses as a function of word frequency may shed more light on how good a model of word ambiguity the Dirichlet Process is.

[9] We note that some POSs in Hungarian have blurred borders, e.g., it is debatable whether the nominal *önkéntes* 'voluntary; volunteer' is ambiguous for its POS.

[10] There are two `mutli` models because Skip-gram and the related MSE models represent each word with two vectors, u and v in the formula $p(w_i \mid w_j) \propto \exp(u_i^\top v_j)$, that `mutli` calls *sense* versus *context* vectors respectively.

	any	disamb
`AdaGram`	73.3%	18.53%
`mutli` sense vectors	71.0%	19.46%
`mutli` context vectors	69.9%	20.76%

Table 4: The resolution trade-off between translation precision and sense distinctiveness. The source models are 600-dimensional Hungarian models trained on the de-glutinized version of the Hungarian National Corpus. Other meta-parameters have been set to default.

Acknowledgements

1957 was an influential year in linguistics: Harris (1957) developed the frequency-aware variant of the distributional method, Osgood et al. (1957) pioneered vector space models, and the author of a more recent conceptual meaning representation framework (Kornai 2010; 2018) was born. Fifty years later (more precisely in fall 2006) I met András during a class he taught on the book he was writing (Kornai 2007). I heard about *deep cases* and *kārakas* sooner than I did about *thematic roles*. He has since taught me computational linguistics and mathematical linguistics in a master and disciple fashion.

Laozi says that a good leader does not leave a footprint, and András encouraged us to be independent and effective. One of his remarkable citations is that "It's easier to ask forgiveness than it is to get permission". The proverb is sometimes attributed to the Jesuits, who are similar to András in having had a great impact on what I've become in the past ten years. The real source of the proverb is Grace Hopper, a US navy admiral who invented the first compiler. This paper is a step in my learning to be so effective as the sources mentioned above.

András Kornai, besides the work already acknowledged, rated each item in Table 3. I would like to thank the anonymous reviewer for detailed critique, both substantial and linguistic, Mátyás Lagos for reviewing language errors, and Gábor Recski and Bálint Sass for their useful comments. The orthogonal approximation was implemented following a code[11] by Gábor Borbély. Veronika Lipp's contribution is section 1.1.

References

Ács, J., K. Pajkossy and A. Kornai. 2013. Building basic vocabulary across 40 languages. In Proceedings of the Sixth Workshop on Building and Using Comparable Corpora. Sofia: Association for Computational Linguistics. 52–58.

Apresjan, J. D. 1974. Regular polysemy. Linguistics 12. 5–32.

Artetxe, M., G. Labaka and E. Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing.

Bartunov, S., D. Kondrashkin, A. Osokin and D. Vetrov. 2016. Breaking sticks and ambiguities with adaptive skip-gram. Proceedings of Machine Learning Research 51. 130–138.

[11] https://github.com/hlt-bme-hu/eval-embed

Borbély, G., A. Kornai, D. Nemeskey and M. Kracht. 2016. Denoising composition in distributional semantics. Poster presented at DSALT: Distributional Semantics and Linguistic Theory.

Cruse, D. A. 2004. Meaning in language. Oxford: Oxford University Press.

Dinu, G., A. Lazaridou and M. Baroni. 2015. Improving zero-shot learning by mitigating the hubness problem. ICLR 2015, Workshop Track.

Faruqui, M. and C. Dyer. 2014. Improving vector space word representations using multilingual correlation. In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics. 462–471.

Halácsy, P., A. Kornai, L. Németh, A. Rung, I. Szakadát and V. Trón. 2004. Creating open language resources for Hungarian. In M. T. Lino, M. F. Xavier, F. Ferreira, R. Costa and R. Silva (eds.) Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC2004) 2004. Paris: European Language Resource Association. 203–210.

Harris, Z. S. 1957. Co-occurrence and transformation in linguistic structure. Language 33. 283–340.

Huang, E., R. Socher, C. Manning and A. Ng. 2012. Improving word representations via global context and multiple word prototypes. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012). Jeju Island, Korea: Association for Computational Linguistics. 873–882.

Jackendoff, R. 2002. Foundations of language: Brain, meaning, grammar, evolution. Oxford: Oxford University Press.

Kornai, A. 2007. Mathematical linguistics. Dordrecht: Springer.

Kornai, A. 2010. The algebra of lexical semantics. In C. Ebert, G. Jäger and J. Michaelis (eds.) The mathematics of language. 10th and 11th Biennial Conference, MOL 10, Los Angeles, CA, USA, July 28–30, 2007 and MOL 11, Bielefeld, Germany, August 20–21, 2009, Revised Selected Papers. Berlin & Heidelberg: Springer. 174–199.

Kornai, A. 2018. Semantics. Berlin: Springer.

Kornai, A., J. Ács, M. Makrai, D. M. Nemeskey, K. Pajkossy and G. Recski. 2015. Competence in lexical semantics. In Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics (*SEM 2015). Denver: Association for Computational Linguistics. 165–175.

Lazaridou, A., G. Dinu and M. Baroni. 2015. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long papers). Association for Computational Linguistics. 270–280.

Li, J. and D. Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding? In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics. 1722–1732.

Lyons, J. 1977. Semantics. Cambridge: Cambridge University Press.

Makrai, M. in preparation. Inter-lingual meaning representations. Doctoral dissertation. Eötvös Loránd University, Budapest.

Mikolov, T., K. Chen, G. Corrado and J. Dean. 2013a. Efficient estimation of word representations in vector space. International Conference on Learning Representations (ICLR 2013).

Mikolov, T., Q. V. Le and I. Sutskever. 2013b. Exploiting similarities among languages for machine translation. Manuscript. https://arxiv.org/abs/1309.4168

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado and J. Dean. 2013c. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani and K. Q. Weinberger (eds.) Advances in neural information processing systems 26. Red Hook, NY: Curran Associates. 3111–3119.

Mikolov, T., W. Yih and G. Zweig. 2013d. Linguistic regularities in continuous space word representations. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013). Atlanta: Association for Computational Linguistics. 746–751.

Murphy, G. L. 2002. The big book of concepts. Cambridge, MA: MIT Press.

Neelakantan, A., J. Shankar, A. Passos and A. McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics. 1059–1069.

Nemeskey, D. M. 2017. emLam – a Hungarian language modeling baseline. In V. Vince (ed.) Konferenciakötet: XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2017), Szeged, 2017. január 26–27. Szeged: Szegedi Tudományegyetem, Informatikai Intézet. 91–102.

Oravecz, C., T. Váradi and B. Sass. 2014. The Hungarian Gigaword Corpus. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014). European Language Resources Association (ELRA).

Osgood, C. E., G. J. Suci and P. H. Tannenbaum. 1957. The measurement of meaning. Urbana, IL: University of Illinois Press.

Pennington, J., R. Socher and C. Manning. 2014. Glove: Global vectors for word representation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014). Association for Computational Linguistics. 1532–1543.

Pustejovsky, J. 1995. The generative lexicon. Cambridge, MA: MIT Press.

Radovanović, M., A. Nanopoulos and M. Ivanović. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. Journal of Machine Learning Research 11. 2487–2531.

Reisinger, J. and R. J. Mooney. 2010. Multi-prototype vector-space models of word meaning. In Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. 109–117.

Schütze, H. 1998. Automatic word sense discrimination. Computational linguistics 24.

Suzuki, I., K. Hara, M. Shimbo, M. Saerens and K. Fukumizu. 2013. Centering similarity measures to reduce hubs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics. 613–623.

Tomašev, N. and D. Mladenic. 2013. Hub co-occurrence modeling for robust high-dimensional knn classification. In H. Blockeel, K. Kersting, S. Nijssen and F. Zelezny (eds.) Machine learning and knowledge discovery in databases. European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23–27, 2013, Proceedings. Berlin: Springer. 643–659.

Xing, C., C. Liu, D. Wang and Y. Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In Proceedings of NAACL. 1005–1010.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 399–424

Hungarian *ugye* is a tag, isn't it?

CECÍLIA SAROLTA MOLNÁR
Research Institute for Linguistics,
Hungarian Academy of Sciences
molnar.cecilia@nytud.mta.hu

KEYWORDS

bias
corpus study
discourse particle
speech act
tag question

ABSTRACT

The paper deals with the formal and functional properties of the Hungarian particle *ugye* and its use in sentences encoding question acts. The investigation is based on a corpus study of the "Budapest Sociolinguistic Interviews". As *ugye* is referred to as a tag, a comparison is made between *ugye*-sentences encoding question acts and English tag questions. This reveals that these constructions share most formal (e.g., basic structure, complex sentence type, resistance to embedding, intolerance to NPIs) and functional properties (e.g., bias for one of the answers, encoding of a complex speech act), although a few differences are also found (e.g., preference of particles, occurrence in declaratives).

I am grateful to László Kálmán for many things. First, and most importantly, I would like to thank him for teaching me to be suspicious of rash theoretical generalizations and respect the diversity and variability of linguistic data.

1. Introduction

The Hungarian discourse particle *ugye* has been investigated by several linguists during the past decades. The process of its development is well known: the elliptical interrogative matrix clause *úgy van-e* 'so be-E' (literally: 'is that so?') was reduced to the shorter form *ugye,* and at the same time, its distribution became less constrained. In contemporary Hungarian it appears to be compatible both with polar interrogatives and declaratives. (These assumptions, however, will have to be qualified later.) As discussed below, there is little agreement in the literature as to how the interpretation of this constituent should be described. The following examples illustrate the variety of its uses in utterances realizing question acts. I am referring to this construction as *ugye*-Q:[1]

(1) (B7313)

I: És számolni, hát számolni azt meg tudni kell.
and count.INF so count.INF that.ACC and know.INF must
'One should definitely know how to count.'

F: **Ugye, hogy tudni kell?**
UGYE that know.INF must
'One should know, shouldn't one?'

I: Nagyon kell, ...
really must
'Of course one should, ...'

(2) (B7402)

F: **Ott magyarul beszéltek, ugye?**
there Hungarian.in spoke.3PL UGYE
'You spoke Hungarian there, right?'

I: Hát ott magyarul beszéltünk,
well there Hungarian.in spoke.1PL
de az első-második osztályban németül tanultunk, ...
but the first-second class.in German.in learned.1PL
'Yes, there we spoke Hungarian, but in the first and the second classes we learned German ...'

(3) (B7307)

LF: **Akkor ugye nem, nem érezte magát ilyen veszélyben, ugye?** ...
then UGYE not not felt.3SG self.ACC such danger.in UGYE
'So, you didn't feel you were in such danger then, did you?'

I (laughing): Nem, nem. Ilyenre nem.
no no such.onto not
'No, no, I didn't.'

[1] Most of my examples are from the corpus of the Budapest Sociolinguistic Interviews (BuSI). In the examples cited here, I did not retain the transcription used in the corpus, I rather follow Hungarian orthography. "I" stands for "informant", and "F" stands for "field worker". Each of the examples is cited together with the number of the interview in which it appears.

(4) (B7404)

LF: **Ugye akkor hol dogozott az édesapja?**
UGYE then where worked.3SG the father.your
'Remind me, where did your father work at that time?'

I: Malomszerelő Vállalatnál dolgozott.
mill-construction company.at worked.3SG
'At the mill construction company.'

This paper investigates the form and the uses of the particle *ugye* in *ugye*-Qs. I use the BuSI corpus as an empirical basis for this investigation, where 239 discourse segments can be found in which *ugye* has the relevant function. The main goal of the paper is to investigate the similarities and the differences between Hungarian *ugye*-Qs and English tag question constructions (TQ). If *ugye*-Qs turn out to be similar to TQs in most respects, their discourse-semantic description should also follow that of TQs. This may be the theoretical impact of my work.[2] The paper is organized as follows. Section 2 briefly reviews earlier descriptive and theoretical work about the contributions of the particle *ugye* in questions. Section 3 presents a possible distinction between tag questions, following Reese's (2007) dissertation. Section 4 contrasts the theoretical generalizations about TQs with Hungarian *ugye*-Q data. Finally, section 5 summarizes the conclusions.

2. Hungarian *ugye* as a question tag

In this section I briefly summarize the main claims of the previous literature on the syntactic distribution and the different uses of the particle *ugye*. In the first part, I go through the descriptive works, and then, in the second part, I discuss semantic-pragmatic analyses that treat *ugye* as a discourse particle. My aim here is to enumerate the main claims on *ugye*, the evaluation or critique of earlier approaches is not in the focus of my work.

[2] Despite the fact that there are many results for particle *ugye* in declarative, and a few in imperative sentences in the BuSI corpus, this article cannot deal with all these uses. Further research should investigate whether these different uses are connected to each other, and if so, how this connection can be described in a coherent way on the level of sentence types, conventional meaning, and discourse function.

2.1. Etymology and descriptive grammars

According to the Hungarian Historical-Etymological Dictionary (Benkő 1967–1984), the first occurrence of the *ugye* particle dates back to 1585. The matrix interrogative clause *úgy van-e* became a compound consisting of the adverb[3] *úgy* 'so', and the polar interrogative marker *-e*. This compound was used in utterances where the speaker intended to confirm, acknowledge, or reinforce the truth of a statement. Thus, the particle was first only used in utterances realizing question acts, where, according to Benkő's assumptions, it had a typical interrogative prosody (a rise-fall contour on the penultimate syllable).[4] Benkő, incorrectly, claims that later the rise-fall contour disappeared, and *ugye* became an "intensifying modifier", more recently a "meaningless expletive element" (*ibid.*, 1027).[5] From the above assumption it would follow that the conditions under which *ugye* can be used currently are less constrained than they were earlier. However, the corpus study below does not prove that the particle can be used freely, without any syntactic constraints as "a meaningless expletive element". I will show that the distribution of *ugye* is constrained by syntactic, semantic, and pragmatic factors.

Kenesei et al.'s (1998) descriptive grammar argues that sentences containing *ugye* encode "leading questions". By using this type of question, the speaker expects agreement or confirmation from the partner. The particle can appear in any syntactic position within the sentence, there is no limitation on its use either in affirmative or in negated sentences. They treat *ugye* as the only marker of biased (or leading) questions.[6] The complementary distribution of the *-e* interrogative particle and *ugye*, which they point out, can be seen as a consequence of this functional differentiation, i.e., *-e* is the marker of neutral questions (cf. Gyuris 2017), while *ugye* is the marker of biased questions. (See also H. Molnár 1959 and Kugler 1998

[3] In Hungarian, Benkő (1967–1984) uses the term *módosítószó* 'modifier word'.

[4] In fact, Benkő (1967–1984) uses the term "interrogative sentence" here, which I think is problematic.

[5] In the course of sketching the historical development of *ugye*, Benkő ignores the fact that in contemporary Hungarian it depends on the intended speech act (question or assertion) whether *ugye* bears the rise-fall intonation contour.

[6] In Hungarian, polar interrogatives are either marked by intonation (rise-fall contour) or by the *-e* particle. Although the use of the latter in root clauses is limited in some dialects, it is acceptable in formal style (e.g., marriage ceremony, legal contexts) for speakers of every dialect. Embedded polar interrogatives are obligatorily marked by the *-e* particle.

who treat *ugye* as the marker of the interrogative sentence type). They do not mention that the particle can also appear in declaratives encoding assertions.

The descriptive grammar of Keszler (2000) claims that the function of *ugye* is similar to that of other "mood markers" (like *-e*, or the interrogative rise-fall intonation), adding that *ugye* usually appears in tag questions, but it does not specify other syntactic environments where the particle can appear. It is also claimed that with *ugye,* the speaker *post factum* modifies the mood of a declarative sentence (which has declarative intonation and expresses a proposition). Note that the fact that the distribution of *ugye* and that of *vajon* (to be discussed below), on the one hand, and the distribution of *ugye* and that of *-e,* on the other hand, are complementary, does not necessary mean that their functions are identical. This can easily be proven by the fact that in a given discourse an *ugye-Q* usually cannot be replaced either by an interrogative sentence containing *-e* or by an interrogative sentence containing *vajon.*

The descriptive syntax of Kálmán (2001) does not mention sentences containing *ugye* in the chapter on questions (*ibid.*, 98–135).[7] This may be due to the fact that *ugye*-Qs fail the syntactic tests of polar interrogatives. According to these, first, a Hungarian sentence is an interrogative if and only if the particle *vajon* can be inserted into it. Second, an interrogative is a polar one in case it can be answered by a simple *nem* 'no' (*ibid.*, 100). The latter criterion aims to differentiate polar interrogatives from *wh*-interrogatives, which cannot be answered by a simple *nem* 'no' in any circumstances.[8] As mentioned above, the distribution of *ugye* and *vajon* is complementary, thus, according to this test, *ugye*-Qs cannot be treated as interrogatives, as (5a) illustrates. However, as (5b), an example from the BuSi corpus shows, *ugye*-Qs (at least with negation) can be answered felicitously by a simple *nem* 'no'. As (5c) illustrates, though, there are cases when a simple *nem* 'no' does not sound like a sufficient answer to *ugye*-Qs, especially in cases where the question has a positive (affirmative) root (or achor) (see also 3.2.).

[7] The title of the chapter is "Questions" (not "interrogatives") despite the fact that it deals mostly with the formal properties of the relevant sentences.

[8] Kiefer (1980) lists several types of polar interrogatives in the case of which a simple *nem* 'no', or *igen* 'yes' answer, although formally adequate, does not sound sufficient or natural.

(5) a. *Vajon ott magyarul beszéltek, ugye?
VAJON there Hungarian.in spoke.3PL UGYE
'You spoke Hungarian there, didn't you?'

b. (B7514)

F: És ugye nem volt azért az olyan borzasztó?
And UGYE not was still that so awful
'It was not so awful, was it?'

I: Nem.
'No.'

c. A: Ott magyarul beszéltek, ugye?
there Hungarian.in spoke.3PL UGYE
'There, you spoke Hungarian, right?'

B: #Nem.
'No.'

B′: Nem, ott már nem magyarul beszéltünk, hanem németül.
no there already not Hungarian.in spoke.1PL but German.in
'No, we didn't speak Hungarian there any more, we spoke German.'

Following Kálmán (2001), we can conclude that *ugye*-Qs are ambivalent in nature: they fail the *vajon*-test, so they are not "real" interrogatives, but, at the same time, they pass the *nem*-as-answer-test.[9] We will see in sections 2.2. and 4.2.2. that other tests also point to the conclusion that *ugye*-Qs do not belong to the interrogative sentence type. Along with this, their semantics/pragmatics is more complicated. We will see that in most cases an *ugye*-Q definitely requires an answer from the partner. The answer can be either *igen* 'yes' or *nem* 'no', and the *ugye*-Q is biased for one of these answers (see 4.2.). In 4.2.3. I argue that in spite of the fact that the particle seems to attach to declaratives, *ugye*-Qs realize question acts.

2.2. Hungarian *ugye* as a discourse particle

There are several recent theoretical and empirical approaches to discourse particles[10] in Hungarian, which also address *ugye*. Gyuris (2008; 2009; 2018) and Alberti and Kleiber (2014) intend to give unified accounts of

[9] Note that a simple *nem* 'no' is a felicitous reaction not only to polar interrrogatives but also to declaratives, thus, the relevant test does not discriminate between the latter two sentence types.

[10] Alternative terms in the literature include that of "discourse marker" or "pragmatic marker".

the distribution and/or interpretation of different uses of *ugye* based on theories of biased questions, or the theory of "context markers". The assumption that there should be a limited number of general rules governing the distribution and interpretation of *ugye* can also be supported by arguments from language acquisition. Gyuris (2009) makes a distinction between two forms: *ugye-declaratives* and *ugye*-sentences encoding a question. She describes the meaning of *ugye* in declaratives by saying that it indicates that the propositional content p of the declarative sentence in which *ugye* appears follows from the Common Ground (*CG*) by default reasoning (following Zeevat 2003). Gyuris (2009) considers *ugye*-questions to be similar to tag questions in English both in their distribution and interpretation, which she judges to be feasible for the following reasons. First, the distributions of particle -*e* and *ugye* are complementary, cf. (6). Second, the distribution of *ugye*-questions and polar interrogatives is not identical: *ugye*-questions cannot be embedded; an embedded *ugye*-sentence can only be interpreted as a declarative, cf. (7a–b).[11] Third, whereas polar interrogatives are compatible with weak NPIs (e.g., *valaha is* 'ever') *ugye*-questions are not, cf. (8). Fourth, the historical development of *ugye* (see 2.1.) and the fact that it first appeared on the peripheries of the clause also points to the conclusion that *ugye* is a tag-like element.

(6) (*Ugye) Mari (*ugye) volt-e (*ugye) Párizsban (*ugye)? (Gyuris 2009, (16))
UGYE Mari UGYE was-E UGYE Paris.in UGYE
'Has Mary been to Paris?'

(7) a. Józsi tudja, hogy Mari ugye volt Párizsban. (*ibid.*, (18))
Józsi knows that Mari UGYE was Paris.in
'Joe knows that, as you know, Mary has been to Paris.'

b. *Józsi tudja, hogy Mari ugye volt-e Párizsban.

(8) *Mari ugye volt valaha is Párizsban? (*ibid.*, (20))
Mari UGYE was ever too Paris.in

Gyuris (2018) derives the interpretation of *ugye* in declaratives from its original interpretation in questions, and provides a unified meaning for the two, according to which *ugye* introduces a condition on input contexts: the interlocutor of the default perspective center of the speech act under consideration (that is, the hearer in assertions and the speaker in questions) is committed to the propositional content.

[11] I will return to this observation in sections 3.1. and 4.2.

Alberti and Kleiber (2014) treat *ugye* as a particle whose "pragmatico-semantic" contribution is to encode the speaker's bias towards the positive answer in "polar interrogatives".[12] Thus, they treat *ugye* as if it had only one function, they ignore its uses in declaratives and other sentence types.

Schirm (2009) presents an empirical study of a corpus of parliamentary discourses. In this corpus *ugye* turned out to be the second most frequent particle after *hát* 'so'. She claims that in declaratives, *ugye* serves to confirm or emphasize, as a default, that a statement is correct/acceptable/right, while in interrogatives it expresses that the speaker expects the positive answer (*ibid.*, 172). The corpus data shows that in parliamentary speech *ugye* has various additional functions. Its use is frequent in emotional, emphatic questions: it indicates that the speaker is happy about some negative developments involving the hearer, or that she blames the latter for some developments.

In parliamentary dialogues *ugye*-Q is often used as a means of argumentation: it encodes a "rhetorical question", by which Schirm means those that cannot be answered, or for which the answer is so obvious that there is no need to formulate it explicitly. The repetitive use of *ugye* enhances the rhetoricity of the text. In addition, it can be seen as a device of self-protection in the case of face-threatening acts: asking a question in general, even an *ugye*-Q, is much less face-threatening than asserting the corresponding proposition (*ibid.*, 173). Summarizing all these features, Schirm claims that *ugye,* generally speaking, expresses the speaker's attitude. But it does not seem easy to identify the contribution of *ugye*, because the sentences cited from the corpus remain rhetorical, and "emotionally loaded" even if we leave out the particle. The question of how these different uses are interconnected also remains open in this work.

Abuczki (2015) works with the most recent Hungarian multi-modal corpus, HuComTech. Based on the corpus data, she identifies three different uses of *ugye:* (i) a tag in tag questions, (ii) an evidence marker or context marker (usually with rhetorical function), (iii) a tool of emphasis, marking new information, truth, explanation, or narrative structure. The possible connections between these interpretations remain unclear.

Despite the number of open questions concerning the different uses of *ugye*, the literature confirms the idea of treating *ugye*-Qs (or at least a subset of them) as tag questions. In the next chapter, I turn to syntactic and semantic properties of English TQs in order to compare them with *ugye*-Qs.

[12] The authors thus disregard the above mentioned difficulties with treating *ugye*-Qs as interrogatives.

3. Tag question constructions in English

Examples (9)–(10) below show that English tag questions (TQs) are complex forms: they consist of a full declarative sentence, the anchor, and a reduced interrogative clause, the tag. Two different types of TQs can be distinguished: (9a,b) are examples of reversed polarity tags, while (10a,b) are examples of constant polarity tags. (The examples are simplified versions of Reese's (2007) examples.)[13]

(9) a. Jane is coming, isn't she?
b. Jane isn't coming, is she?

(10) a. Jane is coming, is she?
b. Jane isn't coming, isn't she?

Compared to positive polar interrogatives, TQs are marked forms. In most uses a TQ is assumed to encode a non-neutral, biased question. In what follows, I am going to summarize Reese's main theses about TQs.

3.1. The form of tag questions

Reese (2007) claims that English TQs are a syntactically mixed sentence type, being composed of a declarative and an interrogative clause. Their structure can be represented schematically as in (11).

(11) [NP Aux (XP)], [Aux Pro] (Reese 2007, 40)

It is easy to see that the form of the tag depends on the form of the anchor. Furthermore, Reese (2007) claims that the form of the tag is constrained by the anchor not only syntactically, but semantically and pragmatically too. The pronoun in the tag must be co-referential with the matrix subject of the anchor. The auxiliary verbs (Aux) used in the anchor and the tag need to be compatible with each other. And the proposition expressed by the anchor need to be a possible answer to the question expressed by the tag.

If we take prosody into account, the above picture about TQs gets more complex. TQs can be pronounced with a falling (12a) or a rising (12b) contour. In addition, following Ladd's fundamental work (Ladd 1981), TQs

[13] According to Reese (2007), the latter type only exists in American dialects, and its use is not widespread. Other authors (e.g., Quirk et al. 1985) do not treat it as a special or rare form.

can be classified as nuclear (13a) or post-nuclear (13b). The latter type is always pronounced with a rising contour, while the former can get both a falling or a rising contour. According to Ladd (1981, 167), nuclear TQs "have a separate nucleus or nuclear pitch accent, generally preceded in the rhythm of the sentence by a noticeable pause or intonational boundary" (indicated by "/"), while post-nuclear TQs "have no separate nucleus, the pitch contour on the tag merely continuing the nuclear contour begun at the preceding nucleus in the main sentence" and "there is noticeably less of a pause or boundary before the tag" (indicated by "="). The possible uses of these forms are also different (see section 3.2.).

(12) a. Jane is coming, isn't she? ↓[14]
b. Jane is coming, isn't she? ↑

(13) a. Jane isn't coming / is she?[15] ↓ / ↑
b. Jane isn't coming = is she? ↑

In the course of investigating the semantics and pragmatics of biased questions, embedding is a useful test for TQs and other marked forms (see Farkas & Roelofsen 2017, 8, examples (15)–(16)).

(14) a. *John told Bill that [Jane is coming, isn't she].
b. *I know that [Jane is coming, isn't she].

The examples in (14) show that English TQs cannot be embedded. For Hungarian *ugye*-Qs the same was shown by Gyuris (2009), see examples (7a–b) of section 2.2. I also applied the "embedding test" for all *ugye*-Q data of the BuSI corpus, the results are presented below, in section 4.2.

3.2. The use of tag questions

Reese (2007) claims that TQs encode complex speech acts: they realize an assertion and a question at the same time.[16] He proves this by applying Sadock's distributional tests (Sadock 1974) to TQs. These tests show that TQs have the distributional properties of both assertions and questions. Sadock assumes that certain discourse markers select utterances with

[14] "↓" marks the *falling* and "↑" the *rising* intonation contour.

[15] I use Ladd's notation for distinguishing nuclear and post-nuclear TQs.

[16] According to other authors (e.g., Farkas & Roelofsen 2017; Malamud & Stephenson 2015; Krifka 2017), the assertion expressed by a TQ is "tentative".

specific illocutionary forces. For example, the expression *after all* can be inserted into sentences that convey an assertion, but not into those that convey a neutral question. Sentences encoding questions (but not assertions), however, remain grammatical after the insertion of the expressions *by any chance* and *tell me*. The latter two expressions can discriminate between neutral and biased questions. *By any chance* can only be inserted into interrogatives encoding neutral questions, while *tell me* is compatible with all types of questions (see Sadock 1974). Applying these tests to TQs, we can see that TQs[17] tend to convey an assertion and a biased question at the same time, but no neutral questions, cf. (15).

(15) a. After all, Jane is coming, isn't she? ↓ (Reese 2007, 51, (13), simplified)
b. #Jane is coming, by any chance, isn't she? ↓ (*ibid.*, 51, (14), simplified)
c. Tell me, Jane is coming, isn't she? ↓ (*ibid.*, 52, (15), simplified)

To sum up, we have seen so far that as a rule, English TQs encode biased questions. Moreover, biased questions conveyed by TQ-forms are differ in interpretation depending on their intonation. On the one hand, if the tag has a falling contour, the speaker is really committed, strongly biased towards the truth of the proposition expressed by the anchor. In this case the function of the TQ is to seek the partner's acknowledgement (*acknowledgement TQ*). On the other hand, if the tag has a rising contour, the speaker has some doubts or uncertainty (or only weak bias) towards the truth of the proposition expressed by the anchor. In this case the function of the TQ is to seek confirmation from the partner (*confirmation TQ*). Thus, both uses are biased, but on a different level.

According to Reese (2007), the strong commitment of the speaker to the anchor of an acknowledgement TQ is of the same type as in an assertion realized by a declarative sentence. As opposed to this, the anchor of a confirmation TQ conveys "weak assertion". As a consequence, a confirmation TQ can be felicitously answered by a plain "no", while an acknowledgement TQ cannot, as (16) shows:

(16) A: Well, that's interesting, isn't it? ↓ (Reese 2007, 58, (24), modified)
B: #No. / #No, it isn't.

Summing up Reese's suggestions about TQs: he says that the semantic/pragmatic complexity of English TQs is due to their complex form,

[17] Reese (2007) presents a special type of TQ that can convey a neutral question: the negative anchor post-nuclear TQ pronounced with rising contour, as in *Jane isn't coming too, by any chance = is she?* ↑ (*op.cit.*, 53, (20b), simplified).

that is, they are of a mixed sentence type. In the next chapter, I turn to Hungarian *ugye*-Qs to compare their properties with those of English TQs.

4. Are Hungarian *ugye*-constructions tag question constructions?

In this section I compare Hungarian *ugye*-Qs with English TQs. I describe the properties of *ugye*-Qs based on my database consisting of BuSI-2 corpus data. In this database I collected all utterances containing the particle *ugye* together with their contexts (together with preceding and following utterances). In what follows, I first briefly present the BuSI-2 corpus and the *ugye*-data in it, and some arguments for using this corpus for this study. Then I turn to the similarities and the differences between Hungarian *ugye*-Qs and English TQs.

4.1. Introductory remarks on the BUSZI-2 corpus

The interviews constituting the BuSI-2 corpus were recorded in 1987 (under the direction of Miklós Kontra). The corpus contains 50 personal interviews conducted by four field workers. Members of five different social groups were involved in the interviews (ten persons in each group): university students, high school teachers, shop assistants, factory workers, and apprentices. For this study, I used the annotated and analyzed transcripts of these anonymous interviews.[18]

The BuSI-2 interviews are not recent, they do not record contemporary spoken language, but it still seemed to be worth using them in the investigation of Hungarian polar interrogative forms (including the problem of *ugye*-Qs). The main argument for using this corpus was that the social statuses of the informants are quite different and they speak in a relaxed, natural manner. Another advantage is that the spoken data is accurately transcribed, and the database can be accessed and searched on-line.[19] One disadvantage is that the 30-year-old recordings are not of a good quality, so for intonational analyses they are inappropriate. Finally, I should admit, that for my purposes it is not ideal that most questions are asked by the field workers.[20]

[18] Despite the fact that I got permission to access some of the sound files of the interviews, I could not properly investigate the intonation pattern of the *ugye*-utterances, because of the bad quality of the recordings.

[19] The BuSI-2 corpus is accessible after a short registration process here: http://buszi.nytud.hu.

[20] Most declarative *ugye*-sentences, however, are produced by the informants.

In BuSI-2 there are 562 dialogues in which *ugye* appears; the number of *ugye*-tokens is higher, because there are dialogues in which it has multiple occurrences. Out of these, 239 utterances realize question acts (*ugye*-Qs). Since BuSI-2 is a spoken language corpus, these utterances are not always realized by complete, grammatical sentences, but there are many (multiply) interrupted, fragmented clauses in it. In most cases (in 217 utterances) the fieldworker asks the *ugye*-Q.

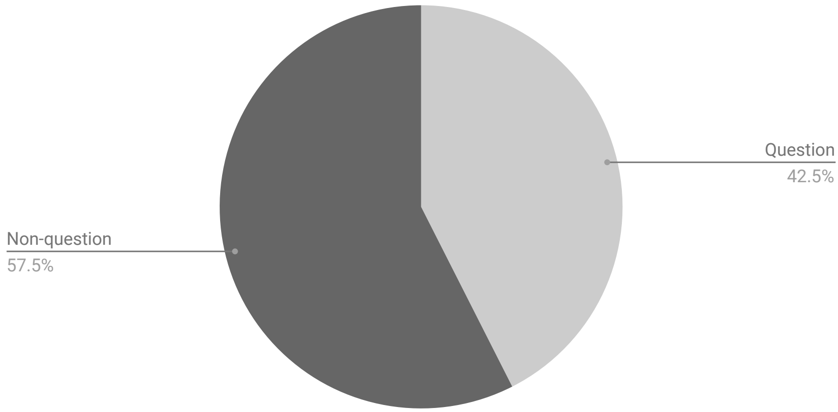


Figure 1: Distribution of *ugye* according to whether the sentences encode a question or not

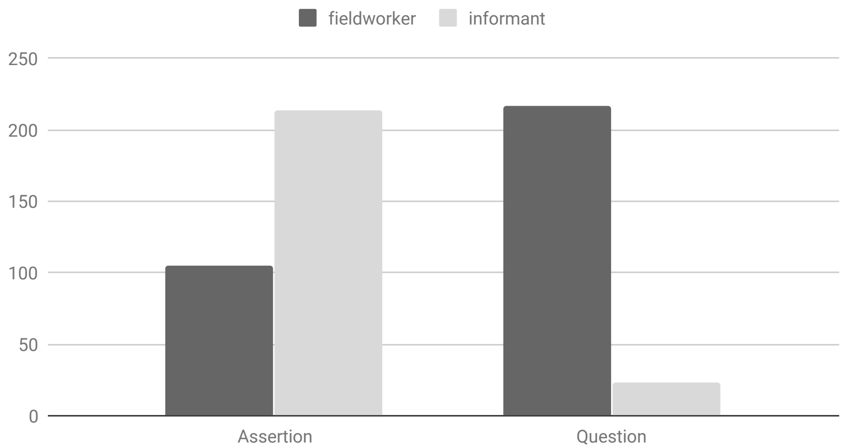


Figure 2: The role of the speakers producing *ugye*-sentences

To distinguish between *ugye*-declaratives realizing only assertions and *ugye*-Qs that can realize questions, I applied the tests proposed for the identification of speech acts by Sadock (1974), discussed above. I categorized an *ugye*-sentence as an *ugye*-Q if it can realize a question act according to the Hungarian counterpart of Sadock's speech act test for questions, i.e., when it remains grammatical after the phrase *mondd csak* 'tell me' is inserted into it (see 4.2. for further discussion).[21] In fact, most of the relevant examples are marked with a question mark (*?*) in the transcription, and the reaction to the utterance could either be *igen* 'yes' or *nem* 'no'.

After having summarized the basic properties of the corpus I used, I turn to the comparison of English TQs and Hungarian *ugye*-Qs.

4.2. Similarities and differences between Hungarian *ugye*-Qs and English TQs

I start the comparison of English TQs and Hungarian *ugye*-Qs with the formal properties of these constructions, and then I turn to their possible functions.

4.2.1. Anchor and tag

According to Keszler (2000) and Gyuris (2009), the forms encoding *ugye*-Qs can be divided into a declarative and an interrogative part, so we can analyze these sentences as consisting of a declarative anchor (the sentence without the particle) and an interrogative tag (the particle itself) – see section 2.

(17) a. (B7301)

F: Szóval maga mindig pesti volt, ugye?
so you always Pest.from was UGYE
'So, you have always been living in Budapest, right?'

I: [Igen]
['Yes']

[21] Note that these utterances also satisfy the Hungarian counterpart of the test proposed by Sadock for the identification of assertions (insertability of the phrase *végül is* 'after all'), see 4.2.3. below for discussion. Thus, if we follow Sadock's and Reese's approach, *ugye*-questions should be assumed to encode both a question and an assertion at the same time.

b. (B7301)

F: Nem tudja, ugye?
not knows UGYE
'You do not know it, do you?'

I: Nem tudom.
not know.1SG
'No, I do not know it.'

In (17a) the anchor is the *Maga mindig pesti volt* 'You have always been living in Budapest' part, to the truth of which the speaker commits herself by uttering the sentence. She adds the tag *ugye* to indicate that she is seeking confirmation from her partner for the truth of the latter. In (17b) the anchor is negated, but the form of the tag remains the same. Thus, while the structure of *ugye*-sentences, and the functions of the "tag" are similar to those of TQs, the form of the Hungarian tag does not depend on the form of the anchor. In many cases *ugye* is interchangeable with other tag-like elements (*nemde?* 'not?', *igaz?* 'right?').

Word order shows another important difference: while in English the tag seems to have a fixed, sentence-final position in most cases, *ugye* can occur in most positions of the Hungarian sentence (see Kenesei et al. 1998). In BuSI-2 there are many examples for sentence-initial as well as non-peripheral occurrences of "questioning" *ugye*.[22]

(18) a. (B7106)

F: Ugye, magának most lukasórája van?
UGYE you.DAT now empty.hour.your be.3SG
'You have free time now, haven't you?'

I: Igen, lukasórám van.
Yes empty.hour.my be.3SG
'Yes, I have free time.'

b. (B7416)

LF: Ott önnel találkoztunk, ugye benn a cégnél?
there you.with met.1PL UGYE inside the company.at
'We once met each other at your company building, right?'

[22] Kenesei et al. (1998) and descriptive grammars (e.g., Keszler 2000) treat *ugye* as completely free element: it is claimed that it can be situated anywhere in the sentence, no word-order constraints delimit its occurrence. Looking more closely at the data, it becomes obvious that its word order is not completely free: it cannot be placed into the immediately pre-verbal position (the so called focus position), for example. Here I cannot go into details about the exact syntactic distribution of the particle.

But if we take the frequency of the word order patterns into account, we can see that examples with sentence-final *ugye* are by far the most frequent (153 occurrences) in BuSI-2, and the second most frequent case is when the particle stands alone (isolated) after a separate declarative sentence (52 occurrences) – see Figure 3.[23] In sum, comparing the ratio of sentences with *ugye* in peripheral positions and those with *ugye* in internal positions we can see that the former case is ten times more frequent than the latter – see Figure 4.[24]

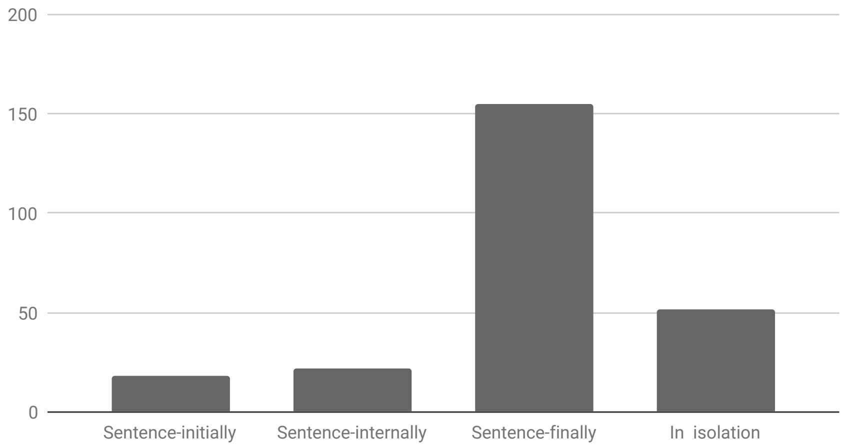


Figure 3: Syntactic distribution of *ugye* in *ugye*-Qs 1

Corpus studies on British and American English presented in Tottie & Hoffmann (2006) show that reversed polarity TQs are significantly more frequent than same polarity TQs in both dialects. Additionally, the positive anchor is much more frequent than the negative one (see *ibid.*, 290, Figure 3). Analysis of the BuSI-2 data revealed the same pattern: negative

[23] I am aware of the problem of using the abstract term "sentence" in case of spoken data, given possible difficulties of segmentation. In my analysis, I consequently relied on the intuition (and the consistency) of the transcriptors. Sentence-final position means that in the transcription there is no full stop after the declarative sentence (the anchor), but there is a full stop after the particle *ugye* (the tag). *Ugye* is treated as "isolated" when there is a full stop after the declarative sentence, and the first letter of *ugye* is capitalized, and there is a question mark after it. (It is transcribed as a separate sentence.)

[24] Although I do not deal with *ugye*-sentences encoding (only) assertions here, I have to mention that in those cases the syntactic distribution of *ugye* is different from that in *ugye*-Qs.

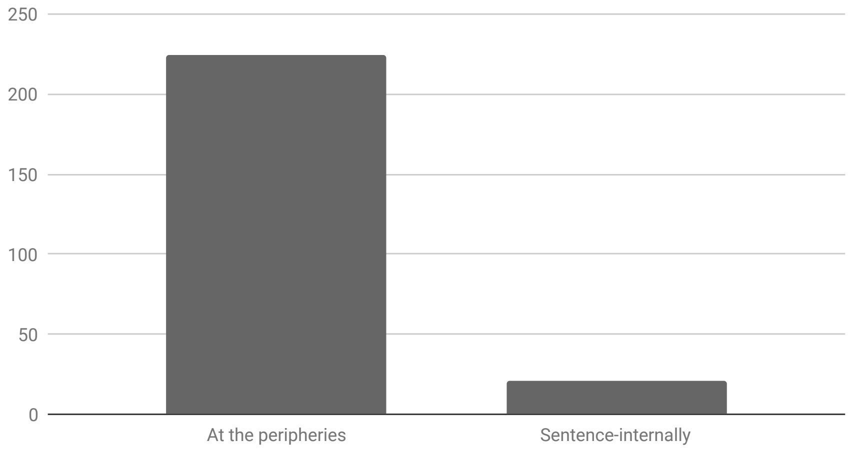


Figure 4: Syntactic distribution of *ugye* in *ugye*-Qs 2

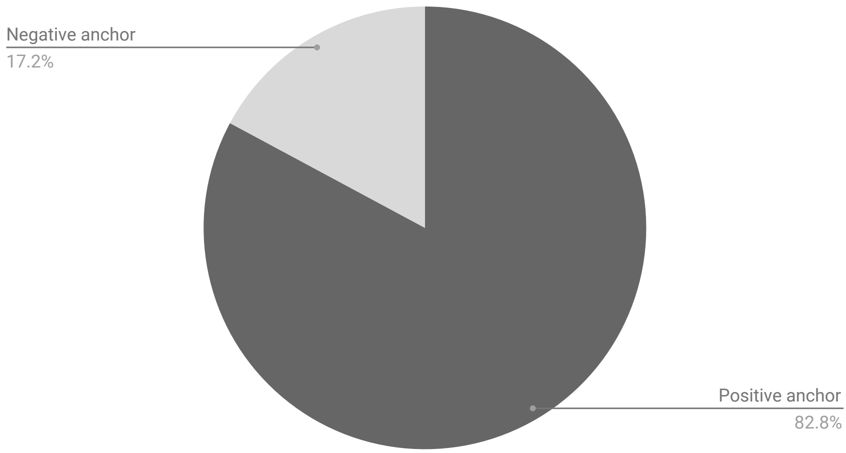


Figure 5: The polarity of the anchors in *ugye*-Qs

anchors are rare in Hungarian as well – Figure 5. (However, I could not find an explanaton for this difference in frequency in the relevant literature.)

As far as syntactic distribution is concerned, we can conclude that although the position of the particle *ugye* within the sentence seems to be relatively unrestricted, it prefers the peripheries of the sentence, especially the right periphery (that is, the sentence-final position). So does the tag in English TQs.

4.2.2. Embedding and negative polarity items

As was already mentioned, Gyuris (2009) presents two tests with which she demonstrates that the distribution of Hungarian polar interrogatives and what she refers to as *ugye*-"interrogatives" (thus avoiding commitment to the interrogative status of structures with *ugye* encoding questions) is not the same (see section 2.2. for further details). One of the tests shows that while canonical polar interrogatives (expressing neutral questions) are grammatical with weak negative polarity items (NPIs), *ugye*-"interrogatives" are not (see example (8) in section 2.2.). Farkas and Roelofsen (2017) made the same observation about English TQs. I applied the test for the utterances of the BuSI-2 corpus I consider *ugye*-Qs, and I found that all of them are ungrammatical with *valaha is* 'ever'. In (20) I show that the insertion of the weak NPI above makes (1)–(2) ungrammatical.

(19) a. *Ugye, hogy valaha is tudni kell?
b. *Ott magyarul beszéltek valaha is, ugye?

The other test by Gyuris (2009) shows that ugye-"interrogatives" cannot be embedded under the matrix declarative *X tudja, hogy...* 'X knows that...'. As mentioned in section 3.1 above, Farkas and Roelofsen (2017) demonstrate the same property for English TQs. Having applied the same test for the *ugy*e-data from BuSI-2, I found that the ones I consider to be *ugye*-Qs cannot be embedded either.

(20) a. *Józsi tudja, hogy ugye, hogy tudni kell.
b. *Józsi tudja, hogy ott magyarul beszéltek, ugye.
c. *Józsi tudja, hogy akkor nem érezte magát ilyen veszélyben, ugye.
'Józsi knows where your father worked at that time.'

Thus, with respect to the weak NPI-insertion and embedding tests, Hungarian *ugye*-Qs and English TQs show the same syntactic behavior, which is different from that of canonical interrogatives encoding neutral questions.

4.2.3. Complex speech act

It was already mentioned in section 3, based on Reese's (2007) assumptions, that the formal complexity of TQs leads to semantic/pragmatic complexity. Applying Sadock's speech act tests (Sadock 1974), Reese shows that English TQs convey complex speech acts: they are assertions and questions at the same time. I applied the speech act tests of Sadock for all *ugye*-data from the BuSi-2 corpus. This means that I tested the grammaticality of each *ugye*-Q with three expressions: (i) *elvégre/végül is* 'after all', (ii) *mondd csak* 'tell me', and (iii) *véletlenül* 'by any chance'. Following Sadock, I assumed that *elvégre* 'after all' can be inserted into sentences encoding assertions, while *mondd csak* 'tell me' can be inserted into sentences encoding questions, and *véletlenül* 'by any chance' can be inserted into sentences encoding unbiased (neutral) questions. The tests showed that almost 80 per cent of the ugye-Qs can be said to realize a question and an assertion at the same time (see Figure 6), and (almost) all *ugye*-Qs are unnatural with *véletlenül* 'by any chance'. Thus, according to Sadock's tests, *ugye*-Qs are not neutral but biased (see example (21)).

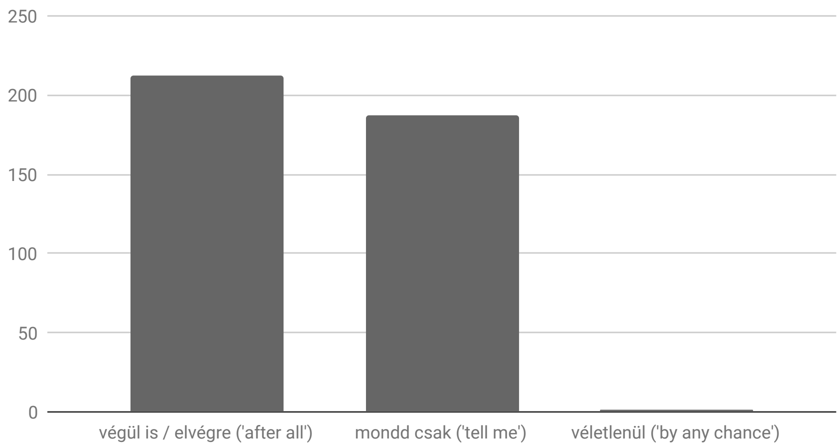


Figure 6: Results of the Sadock-tests for *ugye*-Qs

(21) a. (B7402)
F: Az két év vót, ugye?
that two year was UGYE
'It was two years, wasn't it?'
I: Igen.
'Yes, it was.'

b. Elvégre/végül is az két év volt, ugye?
after.all that two year was UGYE
'After all, it was two years, wasn't it?'

c. Mondd csak, az két év volt, ugye?
tell.IMP.2SG only that two year was UGYE
'Tell me, it was two years, wasn't it?'

d. *Véletlenül az két év volt, ugye?
by.any.chance that two year was UGYE
'By any chance, it was two years, wasn't it?'

Given Figure 6, one can ask what speech act the remaining 20 per cent of the examples realize. As I have already mentioned, analyzing spoken language data is not easy because of the elliptical, interrupted, fragmented structures. *Ugye* is often used in elliptical sentences like (22) and in isolation, like in (23). In these cases the Sadock-tests cannot be applied at all, or can only be applied in a restricted way. This is the reason why only the 80 per cent of the relevant data has turned out to indicate the presence of a complex speech act.

(22) (B7303)

L: Na, utolsó követ(kezik).
so last follow.3SG
'So, this is the last one.'

I: Igen.
'Yes.'

F: Már ideje is, ugye? [laughing] Na, parancsoljon.
already time.its also UGYE so order.IMP.3SG
'It's time for it, right? [laughing] So, here you are...'

(23) (B7105)

F: Hát én is kapocsnak hívom.
well I also brace.DAT call.1SG
'I call it a brace.'

I: Ugye.
'Do you?'

F: Én is kapocsnak hívom.
I also brace.DAT call.1SG
'I would also call it a brace.'

As mentioned in section 3, the discourse function of English TQs depends on their intonation. In case of a falling intonation, the speaker expects acknowledgement form the addressee, and in case of a rising intonation, he/she expects confirmation from the partner. Since I could only use the transcription for the present analysis of the BuSI-2 data, I cannot say anything about these properties of Hungarian *ugye*-Qs. It is worth mentioning, though, that *ugye*-Qs in the corpus hardly ever get negative responses (see Figure 7). Two possible explanations present themselves: (i) *ugye*-Qs only have an acknowledgement reading, and this is why they cannot felicitously be answered by a plain *nem* 'no' without any further explanation, as mentioned above; or (ii) this is a specific characteristic of the BuSI corpus and not of *ugye*-Qs in general, because the participants recorded here were exceptionally polite.

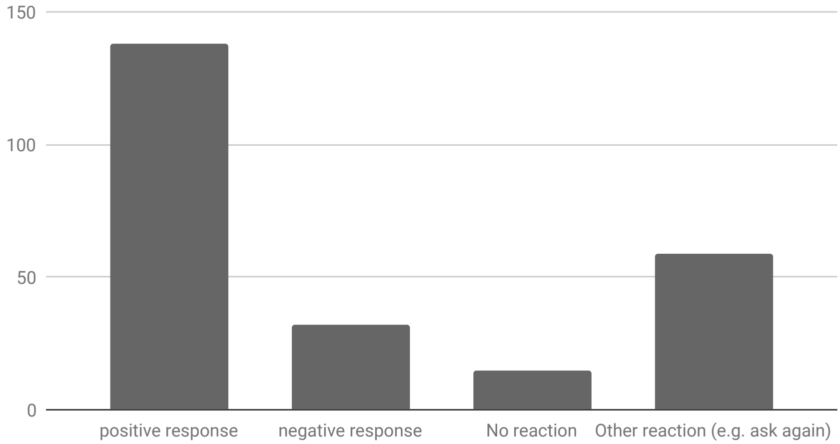


Figure 7: The polarity of the responses for *ugye*-Qs

Examining the polarity of the responses is not enough to make any conclusions about the possibilities of rejecting the propositions expressed by the anchors of *ugye*-Qs. Figure 5 above shows the polarity of *ugye*-Q anchors. Out of 239 situations where *ugye* appears only 18 include an answer rejecting/denying the proposition in the anchor, as in (24).

(24) (B7308)

F: Namost ugye nekem közelebb van itt a villamosmegálló, mint a busz.
so UGYE I.DAT closer is here the tram.stop than the bus
'So the tram station is nearer here than the bus stop, right?'

I: Nem mert itt van lent a buszmegálló.
not because here is down the bus.stop
'No, because the bust stop is just down here.'

F: Ja, igen. Aha.
'Oh, yes, OK.'

So far, we have seen that *ugye*-Qs, like English TQs, are biased. The anchor presents the preferred answer. However, based on the transcription of the

BuSI-2 data alone, we cannot give a conclusive answer about the exact discourse function (seeking confirmation or acknowledgement) of this form.

4.3. Non tag-like properties

I have shown above that in several respects, Hungarian *ugye*-Qs are similar to English TQs. We have also seen that their semantic/pragmatic properties are partly the same. In what follows, I mention some properties of *ugye*-data from BuSI-2 which are not typical tag-like properties. These properties point to the conclusion that we should not categorize *ugye*-Qs as pure TQs.

Unlike English tag elements, *ugye* is not always used as a tag, its use is widespread in declaratives and it appears even in imperative sentences (intended as requests), cf. Figure 1 above.[25] More than half of the *ugye*-tokens appear in declarative sentences in BuSI-2.

As was mentioned above, *ugye* can be used in elliptical sentences and in isolation. This is not typical for English tags (with the probable exception of the invariable tag *innit,* cf. Tottie & Hoffmann 2006).

Another difference between English tag elements and *ugye* is that the latter can appear in constituent questions too, as (4) illustrates. In the BuSI-2 corpus there are 15 results for *ugye* in *wh*-interrogatives. Since *ugye* cannot be considered a tag in constituent questions, the latter use has been assimilated to the use of the particle in declaratives (for further discussion see Molnár 2016).

Finally, Hungarian *ugye* very often co-occurs together with other discourse particles. This seems to be a common main characteristic of all types of *ugye*-sentences. But this is not a property of English tags. Figure 8 shows the particles that co-occur with *ugye* most frequently.

Maybe this property is due to the fact that BuSI-2 interviews come from spontaneous speech, and thus it is not special to *ugye*-sentences.

[25] *Ugye* in sentences with a verb in imperative mood is only illustrated with a few examples in the BuSi corpus, cf. (i) below. (Mood is marked morphologically in Hungarian.)

(i) Na, először is ugye át, azért csak gondosan fusd át,
so first UGYE through though just carefully run.IMP.2SG through
meg nézted végig mind.
VM saw.2SG till.end all
'So first just run through it, just carefully run through it, have you seen it all till the end?'

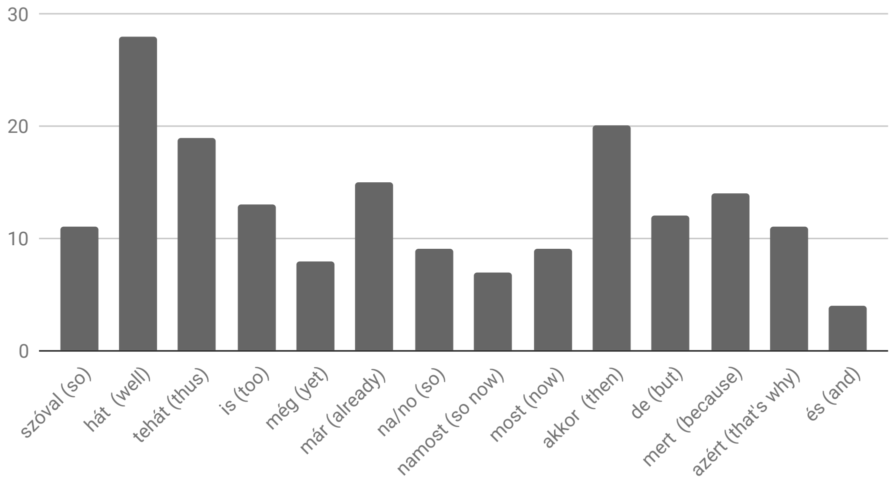


Figure 8: Particles co-occurring with *ugye* in *ugye*-Qs

Figure 9 compares *ugye*-Qs and *ugye*-declaratives with respect to the co-occurrence of *ugye* with other particles.

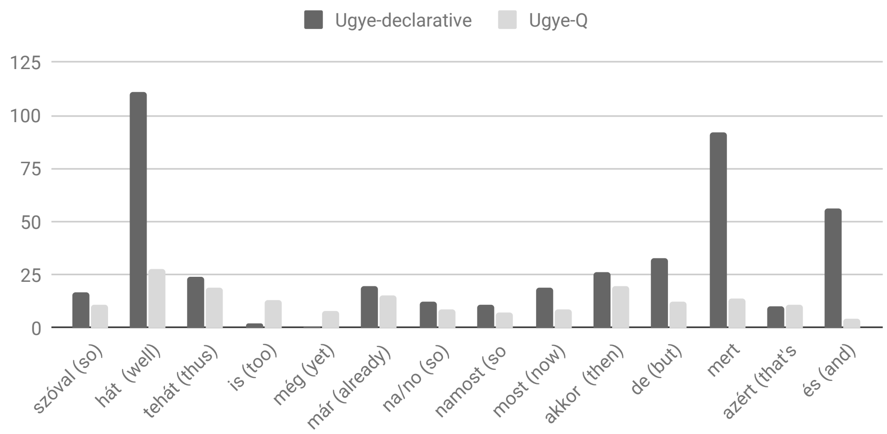


Figure 9: Particles co-occurring with *ugye* in *ugye*-Qs and *ugye*-declaratives

5. Conclusion

This paper has investigated the properties of utterances containing the particle *ugye* that encode question acts, and compared them to those of English TQs. Table 1 summarizes my findings.

Table 1: Properties of English TQs and Hungarian *ugye*-Qs

	English TQs	*Ugye*-Qs
Structure: anchor + tag	+	+
The syntactic position of the tag is fixed	+	–
Tag occurs in elliptical sentences and in isolation	–	+
The proposition expressed by the anchor is a possible answer to the question encoded by the tag	+	+
Co-occurrence with particles is typical	–	+
Multiple occurrences of the tag within one sentence	–	+
Can be embedded under the matrix clause 'X knows that...'	–	–
Co-occurrence with NPIs	–	–
Realizes a complex speech act (assertion + question)	+	+
Encodes a biased question	+	+
The occurrence of the tag is not restricted to utterances realizing question acts	–	+

Acknowledgements

Support by the National Research, Development and Innovation Office – NKFIH, under project no. K 115922 is gratefully acknowledged.

References

Abuczki, Á. 2015. A multimodal discourse-pragmatic analysis of *ugye* ('is that so?'). Sprachtheorie und germanistische Linguistik 25. 41–74.

Alberti, G. and J. Kleiber. 2014. Where are possible worlds? II. Pegs, DRSs, worldlets and reification. In L. Veselovská and M. Janebová (eds.) Complex visibles out there. Proceedings of the Olomouc Linguistics Colloquium 2014: Language Use and Linguistic Structure. Olomouc: Palacký University. 513–528.

Benkő, L. 1967–1984. A magyar nyelv történeti-etimológiai szótára [A historical-etymological dictionary of the Hungarian language]. Budapest: Akadémiai Kiadó.

Farkas, D. and F. Roelofsen. 2017. Division of labor in the interpretation of declaratives and interrogatives. Journal of Semantics 34. 237–289.

Gyuris, B. 2008. A diskurzus-partikulák formális vizsgálata felé [Towards a formal investigation of discourse particles]. In F. Kiefer (ed.) Strukturális magyar nyelvtan 4. A szótár szerkezete [A structural grammar of Hungarian 4. The structure of the lexicon]. Budapest: Akadémiai Kiadó. 639–682.

Gyuris, B. 2009. Sentence-types, discourse particles and intonation in Hungarian. In A. Riester and T. Solstad (eds.) Proceedings of Sinn und Bedeutung 13. Stuttgart: Online Publikationsverbund der Universität Stuttgart (OPUS). 157–171.

Gyuris, B. 2017. New perspectives on bias in polar questions: a study of Hungarian *-e*. International Review of Pragmatics 9. 1–50.

Gyuris, B. 2018. *Ugye* in Hungarian: Towards a unified analysis. In H. Bartos, M. den Dikken, Z. Bánréti and T. Váradi (eds.) Boundaries crossed, at the interfaces of morphosyntax, phonology, pragmatics and semantics. Chem: Springer. 199–212.

H. Molnár, I. 1959. A módosító szók mondattani arculatának kérdéséhez [Towards the syntactic properties of modifying words]. Magyar Nyelv 55. 357–361, 470–480.

Kálmán, L. 2001. Magyar leíró nyelvtan. Mondattan I [Hungarian descriptive grammar. Syntax 1] (Segédkönyvek a nyelvészet tanulmányozásához VI [Supplementary textbooks on linguistics 6]). Budapest: Tinta Könyvkiadó.

Kenesei, I., R. M. Vago and A. Fenyvesi. 1998. Hungarian. London & New York: Routledge.

Keszler, B. (ed.). 2000. Magyar grammatika [Hungarian grammar]. Nemzeti Tankönyvkiadó: Budapest.

Kiefer, F. 1980. Yes-no questions as *wh*-questions. In J. R. Searle, F. Kiefer and M. Bierwisch (eds.) Speech act theory and pragmatics. Dordrecht: D. Reidel. 97–120.

Krifka, M. 2017. Negated polarity questions as denegations of assertions. In C. Lee, F. Kiefer and M. Krifka (eds.) Contrastiveness in information structure, alternatives and scalar implicatures. Cham: Springer. 359–398.

Kugler, N. 1998. A partikula [The particle]. Magyar Nyelvőr 122. 214–219.

Ladd, D. R. 1981. A first look at the semantics and pragmatics of negative questions and tag questions. Proceedings of the Regional Meeting of the Chicago Linguistic Society 17. 164–171.

Malamud, S. A. and T. Stephenson. 2015. Three ways to avoid commitments: Declarative force modifiers in the conversational scoreboard. Journal of Semantics 32. 275–311.

Molnár, C. S. 2016. Ugye melyik szabály alól ne találnánk kivételt? Az *ugye* partikula előfordulása kiegészítendő kérdésekben [The occurrence of the particle *ugye* in interrogative clauses]. Jelentés és Nyelvhasználat 3. 151–167.

Quirk, R., S. Greenbaum, G. Leech and J. Svartvik. 1985. A comprehensive grammar of the English language. London & New York: Longman.

Reese, B. 2007. Bias in questions. Doctoral dissertation. University of Texas at Austin.

Sadock, J. 1974. Towards a linguistic theory of speech acts. New York: Academic Press.

Schirm, A. 2009. Diskurzusjelölők a parlamenti beszédekben [Discourse markers in parliamentary speech]. In K. Kukorelli (ed.) Hatékony nyelvi, idegen nyelvi és szakmai kommunikáció interkulturális környezetben [Efficient linguistic, foreign language and professional communication in intercultural contexts]. Dunaújváros: Dunaújvárosi Főiskola. 168–175.

Tottie, G. and S. Hoffmann. 2006. Tag questions in British and American English. Journal of English Linguistics 34. 283–311.

Zeevat, H. 2003. Particles: Presupposition triggers, context markers or speech act markers. In R. Blutner and H. Zeevat (eds.) Optimality theory and pragmatics. New York: Palgrave Macmillan. 91–111.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 425–443

Indefinite descriptions in typed lambda calculus

ZOLTÁN MOLNÁR
Budapest University of Technology and Economics
mozow@math.bme.hu

KEYWORDS

lambda calculus
epsilon symbol
indefinite descriptions

ABSTRACT

The epsilon calculus seems to be an appropriate environment for modelling the meaning of definite and indefinite descriptions in a natural language. A philosopher of language may ask whether Russell's meaning theory on descriptions is applicable in this language or not. Or more precisely, in what circumstances a sentence (containing an epsilon-expression) has a contextual meaning, and what its logically equivalent quantified reformulation is. The question was answered for first order languages earlier, but the conditions were full of technical complications and the construction applied difficult semantics. In this paper, the question is answered for a typed lambda calculus, in an easier way and by a simpler semantics.

1. Introduction

1.1. Hilbert's epsilon, descriptions and FOL

The first-order language (FOL) extended by the Hilbertian variable binding operator ε is possibly a good choice as an environment modelling a couple of formal linguistic and language philosophical phenomena concerning descriptions.[1] The term

(1) $(\varepsilon x)\varphi$

where φ is a FOL formula and x is a variable, has the following intuitive meaning:

(2) "an F, if there is any F at all"

[1] See Slater (2007); Kneebone (1963, 100).

where predicate F is the intended meaning or the natural language translation of the formula φ. (Here, the notion of translation is due to Tarski. In the present case, the object-language is FOL and the meta-language is the natural language.[2]) The intuition above follows straightforward from the *first epsilon (or transfinite) axiom* introduced by David Hilbert, which is the following formula scheme

$$(\exists x)\varphi(x) \rightarrow \varphi((\varepsilon x)\varphi(x))$$

The intended meaning of the epsilon term shows that $(\varepsilon x)\varphi$ can be called a *conditional indefinite description,* since "an F" alone is an indefinite description, with the addition of the conditional clause "there is any F at all" it becomes a different linguistic entity with, perhaps, a different meaning. Obviously, I do not have to mention that the meaning of the phrase "an F" is itself a problematic one. Therefore, the problem of the semantic difference between "an F" and "an F, if there is any F at all" is also a tricky one. In the paper I am committed to the standpoint that these phrases have the same meaning.

In order to show an application of Hilbert's symbol let me provide a formal reconstruction and analysis of the sentence

(3) The man drinking a martini is interesting-looking.

in FOL extended by ε (this extended language is denoted by FOL+ε).[3] Since, FOL+ε does not contain definite descriptions, the phrase *the man drinking a martini* can be seen as a special case of the use of ε. A possible solution is to add a uniqueness clause to the following formula: "man drinking a martini(x)" (the formula in FOL+ε expressing the natural language predicate *...is a man and drinks a martini*; see Slater 2009, 417):

$$(\varepsilon x)(\textsf{man drinking a martini}(x)\ \&\ (\forall y)(\textsf{man drinking a martini}(y)\ \equiv\ (x = y))$$

Let us denote the term above by

(4) $(\varepsilon_{\mathrm{D}} x)\textsf{man drinking a martini}(x)$

Then sentence (3) is formulated as follows:

(5) $\textsf{interesting-looking}((\varepsilon_{\mathrm{D}} x)(\textsf{man drinking a martini}(x)))$

[2] Cf. the notion of translation as applied in Convention T in (Tarski 1956, 188).

[3] The original sentence can be found in Donnellan (1966).

Let me remark again, that the claim that the phrase *the man drinking a martini* can be expressed by $(\varepsilon_{\mathrm{D}}x)\mathsf{man\ drinking\ a\ martini}(x)$ is not an obvious one, however a possibly good enough working hypothesis. Without a man holding martini in his hand, the meaning of *the man drinking a martini* is as vague as the meaning of the phrase *the man drinking a martini, if anybody at all.*

Accepting the hypothesis above, by sentence (3) one can refer to the interesting-looking person in question, even if he holds a glass of water in his hand. In this case, the semantic value of the term (4) is a person – not drinking a martini – who seems to be interesting.

The problem of sentence (5) reminds one of Russell's Theory of Descriptions (RTD). In Russell's *On Denoting* or in Whitehead & Russell (1910/1967) it is proposed that descriptions must not be treated as proper names, but as incomplete parts of quatified sentences.

> "Thus we must either provide a denotation in cases in which it is at first sight absent, or we must abandon the view that the denotation is what is concerned in propositions which contain denoting phrases." (Russell 1905, 484)

> "According to the view I advocate, a denoting phrase is essentially *part* of a sentence, and does not, as like most single words, have any significance on its own account." (*ibid.*, 488)

According to RTD, a description D, as a denoting phrase, is not interchangeable by an other individual name N which is identical to D, since D is meaningless in separation, and has only contextual meaning. Russell in *On Denoting* (Russell 1905) gives a FOL reformulation for sentences of the form (3), but in the general case, when the natural language sentence contains more than one descriptions or a lot of logical operators the FOL reformulation can be carried out along different lines. One must mind the scope of logical operators and descriptions. Hence, in the general case RTD is appears to be a FOL reformulation program, in the spirit of the treatment of the simple case described in Russell (1905). At this point, a bit naive question arises.

(6) Is the closed formula (5) equivalent to a plain, quantified one?

If it is in general, then RTD, or rather its quantificational program, is applicable to FOL+ε, in the sense that an epsilon-term, containing a closed formula, can be considered as incomplete part of a quantified reformulation.[4] If it is not equivalent in general, then for FOL+ε Donnellan's proposal holds (i.e., sometimes descriptions have separate meaning, too; see Donnellan 1966). The answer to the question seems to be the latter. The term $(\varepsilon x)\varphi$ is a referring one (its semantic value is always defined) and its semantics is unproblematic, even if there is no φ, at least if the reference of $(\varepsilon x)\varphi$ is not a φ. Nevertheless, note that RTD is a strategy proposed to solved the problem 'how to deal with descriptions' and not a (mathematical) thesis. In FOL+ε, $(\varepsilon x)\varphi$ is a proper name (in the sense of Russell), hence the question must be rewritten in a weaker form. But what will be this weakened question?

It is well-known that if the truth value of the sentence $\psi((\varepsilon x)\varphi)$ in any model does not depend on the semantic value of $(\varepsilon x)\varphi$, then there is a FOL reformulation of $\psi((\varepsilon x)\varphi)$.[5] Hence, if $\psi((\varepsilon x)\varphi)$ is an epsilon-invariant formula (its semantic value is independent of the value of the containing epsilon-term) then the term $(\varepsilon x)\varphi$ can be eliminated from $\psi((\varepsilon x)\varphi)$ by a logically equivalent reformulation. The problem is that this plain FOL reformulation is not an explicit or transparent one. The proof of the theorem applies Craig's Interpolation Theorem, which is a pure existence theorem not giving the needed explicit formula. Hence, in the light of the above considerations, the relevant question is the following.

> "Is there an explicit, transparent, well-explainable FOL reformulation of $\psi((\varepsilon x)\varphi)$, provided that $\psi((\varepsilon x)\varphi)$ is epsilon-invariant (in some model)?"

For FOL+ε, the question has been positively answered in Molnár (2013), however with the application of a lot of technical conditions. When one changes FOL to lambda calculus the picture becomes much more clear. The point is that, in FOL the substitution $\psi[x/(\varepsilon x)\varphi]$ is only a meta-language operation, but in the lambda-calculus it is encoded into the object-language via the application MN, where M is an expression of the lambda-language and N is an epsilon-term of the form $(\varepsilon x)P$.

[4] The crucial point in the tradition of RTD is not that what the FOL reformulation is, but wheter there is any such reformulation. For instance, as Zvolenszky puts the question: "Initially, at issue was the meaning of a specific, rather narrow class of expressions, incomplete definite descriptions: are they devices of reference or of quantification?" (Zvolenszky 2007, 1).

[5] One can call it *Caicedo's Theorem* or the *Blass–Gurevich Theorem.* Its proof first presented in Caicedo (1995), but Blass and Gurevich (2000) claim before the theorem (Prop. 3.2.) that the proposition is a folklore and "it is mentioned in Caicedo (1995) without a reference".

1.2. Hilbert's epsilon and the lambda operator

In section 2, a syntax and semantics will be given for the epsilon symbol in the context of typed lambda calculus (TL). The syntactic notions will be the well-known ones, but in the definitions different way will be followed, based on labeled, ordered trees.[6] Since, by the Curry–Howard Correspondence, TL is closely related to the proof theory of the natural deduction system of propositional logic, we make use of the possibility to define the TL notions of TL syntax the same style as proofs. The form of the definitions will fit this doctrine and a tree-based method will be applied.

In section 3, it will be seen that in TL the result can be reached much more faster than in FOL. There is no need to refer to the so-called intensional and substitutional epsilon semantics.[7] The strategy will be the following. The typed lambda language extended by Hilbert's epsilon ($\mathcal{L}_{\lambda}^{\forall\varepsilon}$) will be considered as a formal model of the fragment of the natural language containing descriptions. Then, if it is possible, the epsilon expressions will be eliminated and the sentences containing them will be mapped, in an explicit way, to the epsilon-free quantified reduct $\mathcal{L}_{\lambda}^{\forall}$ of $\mathcal{L}_{\lambda}^{\forall\varepsilon}$. The plain lambda language reformulation will keep the logical truth in the model. Giving Montague-semantics to the extended language and to the plain epsilon-free language as models (the $(\mathfrak{M}, f)$-s and the $\mathfrak{M}$-s below, respectively) the construction will be unproblematic.

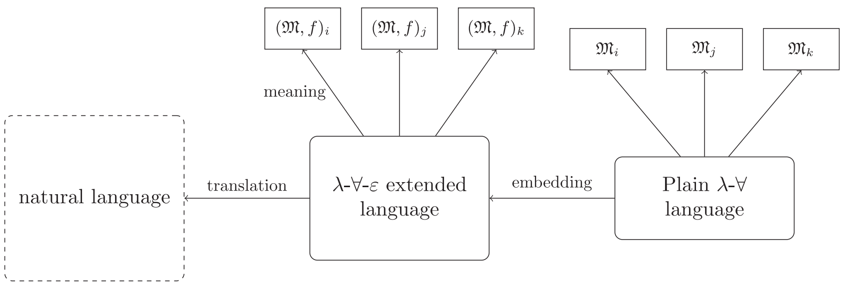


Figure 1: Chain of fragments. The natural language, the λ-∀-ε expressible fragment and the λ-∀ expressible fragment.

[6] It is not easy to refer to a single book or paper, but the book Simmons (2000) (with the programmatic subtitle "Taking the Curry–Howard Correspondence Seriously") surely uses the tree technique that I follow.

[7] Note that, Ahrendt and Giese introduced several types of epsilon semantics. See (Ahrendt & Giese 1999, Def. 4,5). In Molnár (2013) the substitutional semantics was applied. Now, in TL the extensional semantics (see Molnár 2013, 821 or Monk 1976, Def. 29.23) will be enough.

In section 4, it will be pointed out that the result is not less effective than the RTD proposed by Russell.

2. Syntax and semantics of typed lambda system with epsilon

2.1. Syntax

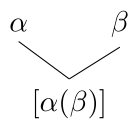
For building the syntax a tree-based method is chosen (parsing or construction trees), which is much more transparent than the old-fashioned character sequence technique. One thing to note is that here the trees grop upward, as those used by by linguists in Combinatory Categorial Grammar, or, what the main motivation is, in proof theory of the style used in natural deduction.

The definitions below are basically combinations of the well-known ones from Troelstra & Schwichtenberg (2000, Sec. 1) and from Sørensen & Urzyczyn (1998).

The so called *typeability relation* ($\vdash$) is a pure syntactic relation that joins the expressions of the lambda calculus to types with respect to a fixed set of typed variables called *context.* Of course, the relation $\vdash$ plays a fundamental role in the Curry–Howard Isomorphism, which links the lambda expressions to proof trees of the natural deduction system of the implicational logic.

Definition 1. The *language of types* is the tuple $\mathcal{L}_{\mathrm{Typ}} = \langle \iota, o, (,), [,] \rangle$. The set of its strings $\mathrm{Str}(\mathcal{L}_{\mathrm{Typ}})$ contains the finite sequences of the characters from $\{\iota, o, (,), [,]\}$. A *construction tree* Π of the string $\gamma \in \mathrm{Str}(\mathcal{L}_{\mathrm{Typ}})$ is a finite, labeled, ordered tree such that the labels of Π are from $\mathrm{Str}(\mathcal{L}_{\mathrm{Typ}})$ and

1. the labels of the leaves of Π come from the set $\{\iota, o\}$,
2. the branch nodes of Π (these are not leaves) and their labels are of the form



3. the root of Π is γ.

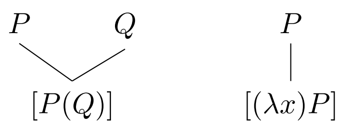
If there is a tree Π such that Π is a construction tree of $\alpha \in \mathrm{Str}(\mathcal{L}_{\mathrm{Typ}})$, then α is said to be a *type (expression)* in $\mathcal{L}_{\mathrm{Typ}}$. The set of all types in $\mathcal{L}_{\mathrm{Typ}}$ is denoted by $\mathrm{Exp}(\mathcal{L}_{\mathrm{Typ}})$. (Cf. Troelstra & Schwichtenberg 2000, Def. 1.2.1 (p. 9); Def. 1.1.7. (p. 7).)

Note that the construction tree of a type is unique. The construction tree of the type α is denoted by $\mathrm{Tree}(\alpha)$. The reference to brackets $[,]$ is avoided when a type α is well-known and its construction tree can be completely reconstructed without them.

Intuitively, ι is the type of individuals and o is the type of sentences. The compound type $o(\iota)$ is, for example, the grammatical type of the single-variable predicates.

Definition 2. A *lambda language* is a tuple $\mathcal{L}_\lambda = \langle V, C, (,), \lambda, [,] \rangle$, where V is an infinite and C is non-empty set and V is disjoint to C. $\mathrm{Str}(\mathcal{L}_\lambda)$ contains the finite sequences from $V \cup C \cup \{\lambda, (,), [,]\}$. A *construction tree* Π of the $M \in \mathrm{Str}(\mathcal{L}_\lambda)$ is a finite, labeled, ordered tree such that the labels of Π are from $\mathrm{Str}(\mathcal{L}_\lambda)$ and

1. the labels of the leaves of Π come from the set $V \cup C$,
2. the branch nodes of Π and their labels are of the form



3. the root of Π is M.

If there is a tree Π such that Π is a construction tree of $M \in \mathrm{Str}(\mathcal{L}_\lambda)$, then M is said to be an *expression* in $\mathcal{L}_\lambda$. The set of all expression in $\mathcal{L}_\lambda$ is denoted by $\mathrm{Exp}(\mathcal{L}_\lambda)$.

The elements of V are called the variables of $\mathcal{L}_\lambda$ and V is denoted by $\mathrm{Var}(\mathcal{L}_\lambda)$. The elements of C are the constants of $\mathcal{L}_\lambda$ and C is denoted by $\mathrm{Const}(\mathcal{L}_\lambda)$. (Cf. Troelstra & Schwichtenberg 2000, Def. 1.2.2 (p. 9); Def. 1.1.7. (p. 7).)

Note that the construction tree of an expression is unique. The construction tree of the expression M is denoted by $\mathrm{Tree}(M)$. The *height* of $\mathrm{Tree}(M)$ is defined by the well-known manner and is denoted by $|\mathrm{Tree}(M)|$.

Referring to brackets $[,]$ is avoided when an expression M is known and its construction tree can be completely reconstructed without them.

Definition 3. Let $\langle V, C, (,), \lambda, [,]\rangle$ be a lambda language. The tuple $\mathcal{L}_\lambda = \langle V, C, (,), \lambda, [,], Z\rangle$ is a *typed lambda language,* if $Z : C \to \mathrm{Exp}(\mathcal{L}_{\mathrm{Typ}})$. The function Z is denoted by $\mathrm{CnstTp}(\mathcal{L}_\lambda)$.

Definition 4. Let $\mathcal{L}_\lambda$ be a lambda language and let $\Xi \subseteq \mathrm{Var}(\mathcal{L}_\lambda)$ be a non-empty finite set. A function $f : \Xi \to \mathrm{Exp}(\mathcal{L}_{\mathrm{Typ}})$ is called a *context,* and the set of all contexts is denoted by $\mathrm{Cont}(\mathcal{L}_\lambda)$. $(\Xi : \Gamma) \in \mathrm{Cont}(\mathcal{L}_\lambda)$ denotes a function f with domain Ξ and range Γ. If $f = (\Xi : \Gamma)$ is a context, and $x \in \Xi$ then $(x : \gamma)$ denotes $f(x) = \gamma$.

For a typed lambda language $\mathcal{L}_\lambda$ the sets of variables, expressions, contexts etc. defined and denoted by the same manner as for a lambda languages.

Definition 5. Let $\mathcal{L}_\lambda$ be a typed lambda language. By induction on the height of the construction tree of the expressions, relation

$$(\Xi : \Gamma) \vdash M : \varphi$$

will be defined as follows for every context $(\Xi : \Gamma) \in \mathrm{Cont}(\mathcal{L}_\lambda)$, expression $M \in \mathrm{Exp}(\mathcal{L}_\lambda)$ and type φ. $\vdash$ is called the *typeability relation.*

1. Let $|\mathrm{Tree}(M)| = 1$.
   a. If $c \in \mathrm{Const}(\mathcal{L}_\lambda)$ and $(\Xi : \Gamma)$ is a context, then $(\Xi : \Gamma) \vdash c : \varphi$, if $\varphi = \mathrm{CnstTp}(\mathcal{L}_\lambda)(c)$.
   b. If $x \in \mathrm{Var}(\mathcal{L}_\lambda)$ and $(\Xi : \Gamma)$ is a context, then $(\Xi : \Gamma) \vdash x : \varphi$, if $(x : \varphi) \in (\Xi : \Gamma)$.
2. Let us suppose that $n > 1$ and for every $(\Upsilon : \Delta)$ context, type ψ and expression N with $|\mathrm{Tree}(N)| < n$, the relation $(\Upsilon : \Delta) \vdash N : \psi$ is defined. Let $(\Xi : \Gamma)$ be a context, φ a type and M an expression such that $|\mathrm{Tree}(M)| = n$.
   a. Let $M = P(Q)$. Then $(\Xi : \Gamma) \vdash M : \varphi$, if $(\Xi : \Gamma) \vdash Q : \beta$ and $(\Xi : \Gamma) \vdash P : \alpha(\beta)$ and $\varphi = \alpha$.
   b. Let $M = (\lambda x)P$. Then $(\Xi : \Gamma) \vdash M : \varphi$, if $(\Upsilon : \Delta) \vdash P : \alpha$, $\varphi = \alpha(\beta)$ and $(\Xi : \Gamma) = (\Upsilon : \Delta) \setminus \{(x : \beta)\}$.[8],

For some examples, see Troelstra & Schwichtenberg (2000, 10).

[8] Cf. Sørensen & Urzyczyn (1998, 41, def. 3.1.1.).

2.2. Montague-semantics

Definition 6. Let $M \neq \emptyset$. By induction on $|\mathrm{Tree}(\varphi)|$, the domain set $D_M(\varphi)$ of type $\varphi \in \mathcal{L}_{\mathrm{Typ}}$ is defined as follows.

1. $D_M(o) = \{\mathsf{T}, \mathsf{F}\}, \qquad D_M(\iota) = M$
2. If $D_M(\alpha)$ and $D_M(\beta)$ is defined earlier, then

$$D_M(\alpha(\beta)) = {}^{D_M(\beta)}D_M(\alpha)$$

where ${}^{D_M(\beta)}D_M(\alpha)$ is the set $\{f : D_M(\beta) \to D_M(\alpha)\}$.

If M is fixed, then $D(\varphi)$ is written instead.

Definition 7. If $M \neq \emptyset$, $\mathcal{L}_\lambda$ is a lambda-language and $(\Xi : \Gamma)$ is a context, then a function $a : \mathrm{Var}(\mathcal{L}_\lambda) \to \cup_{\varphi \in \mathcal{L}_{\mathrm{Typ}}} D_M(\varphi)$ is an *assignation* of the variables. The assignation a is an *assignation of the type* $(\Xi : \Gamma)$, if for every $x \in \Xi$, $a(x) \in D_M(\alpha)$ whenever $(x : \alpha) \in (\Xi : \Gamma)$.

Definition 8. Let $\mathcal{L}_\lambda$ be a typed lambda-language, $M \neq \emptyset$. The tuple $\mathfrak{M} = \langle M, \mathrm{Ip}^{\mathfrak{M}} \rangle$ is a *model* over the language $\mathcal{L}_\lambda$, if $\mathrm{Ip}^{\mathfrak{M}} : C \to \cup_{\varphi \in \mathcal{L}_{\mathrm{Typ}}} D(\varphi)$ such that $\mathrm{Ip}^{\mathfrak{M}}(c) \in D(\mathrm{CnstTp}(\mathcal{L}_\lambda)(c))$.

Definition 9. Let $\mathcal{L}_\lambda$ be a typed lambda-language, $\mathfrak{M} = \langle M, \mathrm{Ip}^{\mathfrak{M}} \rangle$ a model over the language $\mathcal{L}_\lambda$, $(\Xi : \Gamma)$ a context and a an assignation of the type $(\Xi : \Gamma)$. Suppose that for $N \in \mathrm{Exp}(\mathcal{L}_\lambda)$ there is a type φ such that $(\Xi : \Gamma) \vdash N : \varphi$. By induction on $|\mathrm{Tree}(N)|$ the semantic value $[\![N]\!]_a^{\mathfrak{M}}$ in context $(\Xi : \Gamma)$ is defined as follows.

1. If $N = c \in \mathrm{Const}(\mathcal{L}_\lambda)$, then

$$[\![c]\!]_a^{\mathfrak{M}} = \mathrm{Ip}^{\mathfrak{M}}(c).$$

2. If $N = x \in \mathrm{Var}(\mathcal{L}_\lambda)$, then

$$[\![x]\!]_a^{\mathfrak{M}} = a(x).$$

3. Let $N = P(Q)$, then

$$[\![P(Q)]\!]_a^{\mathfrak{M}} = [\![P]\!]_a^{\mathfrak{M}}([\![Q]\!]_a^{\mathfrak{M}}).$$

4. Let $N = (\lambda x)P$ and let the assignment $a[x \to \xi]$ be the following:

$a[x \to \xi](y) = a(y)$ for every variable $y \neq x$, and $a(x) = \xi$.

Then

$$[\![(\lambda x)P]\!]^{\mathfrak{M}}_{a} : D(\alpha) \to D(\beta) \; ; \xi \mapsto [\![P]\!]^{\mathfrak{M}}_{a[x\to\xi]}$$

where $(\Xi : \Gamma) \vdash x : \alpha$ and $(\Xi : \Gamma) \vdash P : \beta$.

Note that if N is not typeable in a context $(\Xi : \Gamma)$, i.e., there is no type φ such that

$$(\Xi : \Gamma) \vdash N : \varphi$$

then N has no semantic value in an assignment of the type of the context. For example, let the type of the constant c be $o(\iota)$ and the context $(\Xi : \Gamma) = \{(x : o)\}$. Then the expression $c(x)$ is not typeable from the context $\{(x : o)\}$, since the argument of c must be an expression of the type ι. However, $c(x)$ is a well-defined expression, it has no semantic value in the context $\{(x : o)\}$.

2.3. Logical and epsilon extensions

The logical operators will be defined as constants of certain types. If $\mathcal{L}_\lambda$ is a typed lambda language, then it could be extended by the following constants.

1. $\neg : o(o)$ $\quad$ $\mathrm{Ip}^{\mathfrak{M}}(\neg) : \mathsf{T} \mapsto \mathsf{F}, \mathsf{F} \mapsto \mathsf{T}$ in a model $\mathfrak{M}$,
2. $\vee : o(o(o))$ $\quad$ $\mathrm{Ip}^{\mathfrak{M}}(\vee) : (\mathsf{F}, \mathsf{F}) \mapsto \mathsf{F},$ and T otherwise in a model $\mathfrak{M}$,
3. $\forall : o(o(\iota))$ $\quad$ $\mathrm{Ip}^{\mathfrak{M}}(\forall) : \{\mathsf{T}, \mathsf{F}\}^M \to \{\mathsf{T}, \mathsf{F}\},\ (M \to \{\mathsf{T}, \mathsf{F}\}; \xi \mapsto \mathsf{T}) \mapsto \mathsf{T},$ and F otherwise in a model $\mathfrak{M}$,
4. $\varepsilon : \iota(o(\iota))$ $\quad$ $\mathrm{Ip}^{\mathfrak{M}}(\varepsilon) : \{\mathsf{T}, \mathsf{F}\}^M \to M : f \mapsto g(\{\xi \in M \mid f(\xi) = \mathsf{T}\}),$ where g is a fixed *choice function* $\mathcal{P}(M) \to M$ such that $g(S) \in S$, if $S \neq \emptyset$ and $g(S) \in M$, if $S = \emptyset$, in a model $\mathfrak{M}$.

In what follows, two specific extensions will be made use of, the *plain extension*

$$\mathcal{L}_\lambda^\forall \text{ with } \mathrm{Const}(\mathcal{L}_\lambda^\forall) = \mathrm{Const}(\mathcal{L}_\lambda) \cup \{\neg, \vee, \forall\}$$

and the *epsilon extension*

$$\mathcal{L}_\lambda^{\forall\varepsilon} \text{ with } \mathrm{Const}(\mathcal{L}_\lambda^{\forall\varepsilon}) = \mathrm{Const}(\mathcal{L}_\lambda) \cup \{\neg, \vee, \forall, \varepsilon\}.$$

If $\mathfrak{M}$ is a model of $\mathcal{L}_\lambda^\forall$, then $(\mathfrak{M}, g)$ will denote the (expanded) model of the $\mathcal{L}_\lambda^{\forall\varepsilon}$ extension with a choice function g described above.[9]

[9] Actually, epsilon-terms are a special kind of Skolem functions; it is pointed out in Monk (1976, 481) and in Mints (1996, sec. 2).

Some further (classical) notations will also be used:

$$P \to Q = \vee([\neg(P)](Q)), P\&Q = \neg(\vee([\neg(P)](\neg(Q)))), (\forall x)P = \forall((\lambda x)P)$$

$$(\varepsilon x)P = \varepsilon((\lambda x)P).$$

For further purposes the language $\mathcal{L}_\lambda^{\forall\varepsilon=}$ using *identity* of individuals is also introduced and the meaning of $=$ is defined as

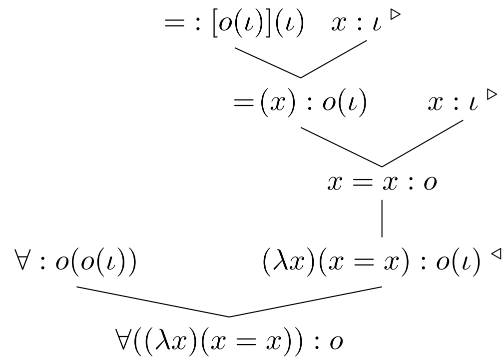
5. $=: (o(\iota))(\iota)$ $\quad \mathrm{Ip}^{\mathfrak{M}}(=) : M^2 \to \{\mathsf{T},\mathsf{F}\}, (x,y) \mapsto \mathsf{T}$, if $x = y$ and F otherwise in a model $\mathfrak{M}$.

2.4. Examples

Proposition 1. Let x be a variable and $(\mathfrak{M}, g)$ be a model over the language $\mathcal{L}_\lambda^{\forall\varepsilon=}$. Then

1. $\vdash (\forall x)(x = x) : o$
2. $\vdash (\varepsilon x)(x \neq x) = (\varepsilon x)(x \neq x) : o$
3. $[\![(\forall x)(x = x)]\!]^{(\mathfrak{M},g)} = [\![(\varepsilon x)(x \neq x) = (\varepsilon x)(x \neq x)]\!]^{(\mathfrak{M},g)} = \mathsf{T}$

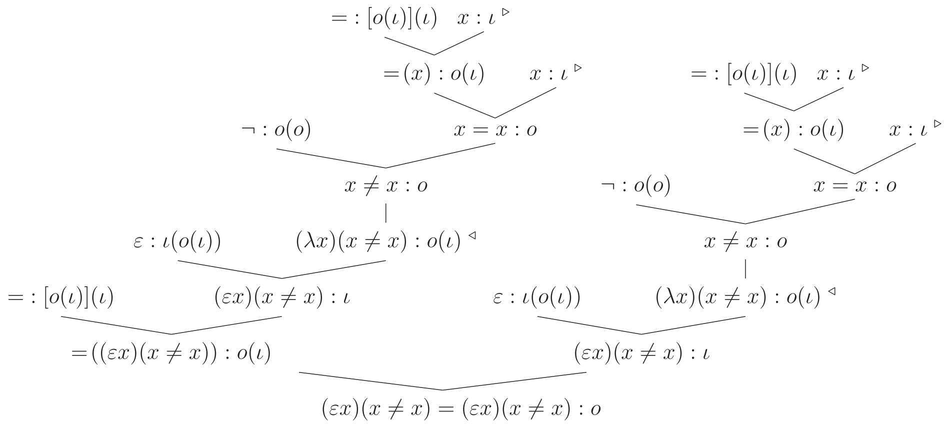
Proof. (1)



Here $(= (x))(x)$ is denoted by $x = x$. The proof tree above shows that the expression $\forall((\lambda x)(x = x))$, which is the same as $(\forall x)(x = x)$, is typeabe by the type o. The labels $^{\triangleright}$ on the left sides of the leaves mark the places which are called the "dischargeable premises" in proof theory. $^{\triangleleft}$ marks the node where they are abandoned. According to part (2b) of Definition 5, both the $x : \iota$-s are discharged by the node $(\lambda x)([= (x)](x)) : o(\iota)$, i.e.,

$(x : \iota)$ can be canceled from the context, which is now an empty set. Note that, the use of the labels $^{\triangleleft}$ and $^{\triangleright}$ is completely unnecessary, since the role of the variable x is exactly that of the triangles. The variable x in the leaves marks the "dischargeable premises" and the symbol (λx) marks the node discharging the premises labeled by the free variable x, after which x becomes a bound variable.

(2)



Here, according to the definitions of ε and $=$ as constants above, $\varepsilon((\lambda x)(x \neq x))$ is denoted by $(\varepsilon x)(x \neq x)$ and $\neg(x = x)$ is denoted by $x \neq x$. The above proof tree proves that $(\varepsilon x)(x \neq x) = (\varepsilon x)(x \neq x)$ is typeable by o.

(3)

$$\begin{aligned}
[\![(\forall x)(x = x)]\!]_a^{(\mathfrak{M},g)} &= [\![\forall((\lambda x)(x = x)]\!]_a^{(\mathfrak{M},g)} \\
&= [\![\forall]\!]_a^{(\mathfrak{M},g)}([\![(\lambda x)(x = x)]\!]^{(\mathfrak{M},g)})_a \\
&= [\![\forall]\!]_a^{(\mathfrak{M},g)}(\xi \mapsto [\![= (x)(x)]\!]_{a[x \to \xi]}^{(\mathfrak{M},g)}) \\
&= [\![\forall]\!]_a^{(\mathfrak{M},g)}(\xi \mapsto [\![= (x)]\!]_{a[x \to \xi]}^{(\mathfrak{M},g)}(\xi)) \\
&= [\![\forall]\!]_a^{(\mathfrak{M},g)}(\xi \mapsto [\![=]\!]_{a[x \to \xi]}^{(\mathfrak{M},g)}(\xi)(\xi)) \\
&= [\![\forall]\!]_a^{(\mathfrak{M},g)}(\xi \mapsto \mathsf{T}) \\
&= \mathsf{T}
\end{aligned}$$

The second expression's semantic value is trivial:

$$\begin{aligned}
[\![(\varepsilon x)(x \neq x) = (\varepsilon x)(x \neq x)]\!]_a^{(\mathfrak{M},g)} &= [\![=]\!]_a^{(\mathfrak{M},g)}([\![(\varepsilon x)(x \neq x)]\!]_a^{(\mathfrak{M},g)})([\![(\varepsilon x)(x \neq x)]\!]_a^{(\mathfrak{M},g)}) \\
&= \mathsf{T}
\end{aligned}$$

below, we determine it:

$$\begin{aligned}
[\![(\varepsilon x)(x \neq x)]\!]_a^{(\mathfrak{M},g)} &= [\![\varepsilon((\lambda x)(x \neq x))]\!]_a^{(\mathfrak{M},g)} \\
&= [\![\varepsilon]\!]_a^{(\mathfrak{M},g)}([\![(\lambda x)(x \neq x)]\!]_a^{(\mathfrak{M},g)}) \\
&= [\![\varepsilon]\!]_a^{(\mathfrak{M},g)}(\xi \mapsto [\![x \neq x]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)}) \\
&= g(\{\xi \in M \mid [\![x \neq x]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)} = \mathsf{T}\}) = g(\emptyset) \\
&= g(\{\xi \in M \mid [\![\neg]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)}([\![=]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)}(\xi)(\xi)) = \mathsf{T}\}) = g(\emptyset)
\end{aligned}$$

□

2.5. Epsilon-invariant expressions

Definition 10. Let $N \in \mathrm{Exp}(\mathcal{L}_\lambda^{\forall\varepsilon})$ be such that for a context $(\Xi : \Gamma)$ the relation $(\Xi : \Gamma) \vdash N : \varphi$ holds for a type φ and let $\mathfrak{M}$ be a $\mathcal{L}_\lambda^{\forall}$ model. N is said to be *epsilon-invariant over the model* $\mathfrak{M}$, if for every assignment a of type $(\Xi : \Gamma)$ and choice functions $g_1, g_2 : \mathcal{P}(M) \to M$ it holds that

$$[\![N]\!]_a^{(\mathfrak{M},g_1)} = [\![N]\!]_a^{(\mathfrak{M},g_2)}.$$

The notion above is a symbolic formulation of the intuitive term "epsilon-independent". In FOL this concept was applied to show that "epsilon-independent" sentences can be reformulated into an epsilon-free one, provided the sentence is independent over *every* model (see Blass & Gurevich 2000).

3. Epsilon and application

Theorem 1. Let $P, Q \in \mathrm{Exp}(\mathcal{L}_\lambda^{\forall\varepsilon})$, $\mathfrak{M}$ be a model of $\mathcal{L}_\lambda^{\forall}$, $(\Xi : \Gamma)$ a context, $(\Xi : \Gamma) \vdash P : o$, $(\Xi : \Gamma) \vdash Q : o$ and $x \in \mathrm{Var}(\mathcal{L}_\lambda^{\forall\varepsilon})$, furthermore, let $[(\lambda x)P]((\varepsilon x)Q)$, P and Q be epsilon-invariant over the model $\mathfrak{M}$. Then for every assignment a of type $(\Xi : \Gamma)$ and choice function $g : \mathcal{P}(M) \to M$:

$$[\![[(\lambda x)P]((\varepsilon x)Q)]\!]_a^{(\mathfrak{M},g)} = [\![((\forall x)(\neg Q)\&(\forall x)P) \vee (((\exists x)Q)\&(\forall x)(Q \to P))]\!]_a^{(\mathfrak{M},g)}.$$

Proof. (1) Let the right hand side be T.
First case: $[\![((\forall x)(\neg Q)\&(\forall x)P)]\!]_a^{(\mathfrak{M},g)} = \mathsf{T}$. Then $[\![(\forall x)P)]\!]_a^{(\mathfrak{M},g)} = \mathsf{T}$ holds and let $m = [\![(\varepsilon x)Q]\!]_a^{(\mathfrak{M},g)} \in M$. Hence, by definition

$$\mathsf{T} = [\![(\forall x)P)]\!]_a^{(\mathfrak{M},g)} = [\![\forall((\lambda x)P)]\!]_a^{(\mathfrak{M},g)}$$

that is

$$[\![(\lambda x)P]\!]_a^{(\mathfrak{M},g)} = \left(\xi \mapsto [\![P]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)}\right) \equiv \mathsf{T}.$$

Hence

$$[\![[(\lambda x)P]((\varepsilon x)Q)]\!]_a^{(\mathfrak{M},g)} = [\![[(\lambda x)P]]\!]_a^{(\mathfrak{M},g)}(m) = [\![P]\!]_{a[x\to m]}^{(\mathfrak{M},g)} = \mathsf{T}.$$

Second case: $[\![(((\exists x)Q)\&(\forall x)(Q \to P))]\!]_a^{(\mathfrak{M},g)} = \mathsf{T}$. Then

$$[\![(\lambda x)\neg Q]\!]_a^{(\mathfrak{M},g)} = \left(\xi \mapsto [\![\neg Q]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)}\right) \not\equiv \mathsf{T}$$

hence for a $\xi \in M$ $[\![Q]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)} = \mathsf{T}$. Therefore, if $[\![\varepsilon((\lambda x)Q)]\!]_a^{(\mathfrak{M},g)} = m$ then $[\![(\lambda x)Q]\!]_a^{(\mathfrak{M},g)}(m) = \mathsf{T}$. But from $[\![(\forall x)(Q \to P))]\!]_a^{(\mathfrak{M},g)} = \mathsf{T}$ it follows that $[\![P]\!]_{a[x\to m]}^{(\mathfrak{M},g)} = \mathsf{T}$, since $[\![Q]\!]_{a[x\to m]}^{(\mathfrak{M},g)} = \mathsf{T}$. Hence, $[\![(\lambda x)P]\!]_a^{(\mathfrak{M},g)}(m) = [\![[(\lambda x)P]((\varepsilon x)Q)]\!]_a^{(\mathfrak{M},g)} = \mathsf{T}$.

(2) Suppose the left hand side is T. First case: let $[\![((\forall x)(\neg Q)]\!]_a^{(\mathfrak{M},g)} = \mathsf{T}$, $m \in M$ arbitrary and g' is the choice function such that $g'(\emptyset) = m$. Hence, by the epsilon-invariance of P and $[(\lambda x)P]((\varepsilon x)Q)$ it follows that

$$\mathsf{T} = [\![[(\lambda x)P]((\varepsilon x)Q)]\!]_a^{(\mathfrak{M},g)} = [\![[(\lambda x)P]((\varepsilon x)Q)]\!]_a^{(\mathfrak{M},g')} = [\![P]\!]_{a[x\to m]}^{(\mathfrak{M},g')} = [\![P]\!]_{a[x\to m]}^{(\mathfrak{M},g)}$$

therefore $[\![(\forall x)P]\!]_a^{(\mathfrak{M},g)} = \mathsf{T}$. Second case: let $[\![((\exists x)Q]\!]_a^{(\mathfrak{M},g)} = \mathsf{T}$, $m \in M$ arbitrary such that $[\![Q]\!]_{a[x\to m]}^{(\mathfrak{M},g)} = \mathsf{T}$ and g' is the choice function such that $g'(\{\xi \in M \mid [\![Q]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)} = \mathsf{T}\}) = m$. Then by the epsilon-invariance of P, Q and $[(\lambda x)P]((\varepsilon x)Q)$ it follows that

$$\mathsf{T} = [\![[(\lambda x)P]((\varepsilon x)Q)]\!]_a^{(\mathfrak{M},g)} = [\![[(\lambda x)P]((\varepsilon x)Q)]\!]_a^{(\mathfrak{M},g')} = [\![P]\!]_{a[x\to m]}^{(\mathfrak{M},g')} = [\![P]\!]_{a[x\to m]}^{(\mathfrak{M},g)}$$

for every m such that $[\![Q]\!]_{a[x\to m]}^{(\mathfrak{M},g)} = \mathsf{T}$. Hence, $[\![(\forall x)(Q \to P)]\!]_a^{(\mathfrak{M},g)} = \mathsf{T}$

□

4. Morning Star and King of France tests

The concluding facts can be stated in two claims:

1. In the formal language $\mathcal{L}_\lambda^{\forall\varepsilon}$ (which is supposed to model the behaviour of descriptions) the (closed) term $(\varepsilon x)Q$ has *referential meaning* in the sense that a fixed model $(\mathfrak{M}, g)$ singles out an individual $[\![(\varepsilon x)Q]\!]^{(\mathfrak{M},g)} \in M$ for $(\varepsilon x)Q$ as semantic value.

2. In some cases, when $(\varepsilon x)Q$ is part of a compound sentence $[(\lambda x)P]((\varepsilon x)Q)$, with all its components being epsilon-invariant, the $(\varepsilon x)Q$ has a *contextual meaning,* such that the sentence $[(\lambda x)P]((\varepsilon x)Q)$ has an equivalent epsilon-free reformulation using quantified expressions from the plain language $\mathcal{L}_\lambda^{\forall}$.

We do not intend to set up a weaker theory than Russell's Theory of Descriptions. A new theory must serve at least as many solutions as far as Russell's proposal was able to solve. An appropriate indicator is to look at the two problems that the Theory of Descriptions solved and examine what the new model porposes. The first one is the problem of Hesperus and Phosphorus (below it will be called Morning Star Test), the second one is the problem of the empty names (the King of France Test).

4.1. Morning Star Test

In 1905, Russell gave a FOL-based solution of the so-called Frege Puzzle in terms of RTD, understandably, without mentioning the intensional tools of possible world semantics, which is a much later development. Here, I would like to show briefly that even the exposition of the puzzle is so widely criticized, that the RTD result of the test is rather irrelevant to us.

> "Gottlob thinks that the Morning Star is illuminated by the Sun."
> "The Evening Star is the Morning Star."
> —
> "Gottlob thinks that the Evening Star is illuminated by the Sun".
> (Cf. Frege 1892/1990.)

First of all, I would like to point out that several scholars are committed to the assumption that the names such as *the Morning Star* or *the Evening Star* are understood tacitly as definite descriptions. For Russell, these names abbreviate descriptions, hence they are denoting phrases too (Dummett 1973, 97). The problem is that, according to Leiniz's Rule, since the Evening Star is the Morning Star, the two phrases are interchangeable. However, the above inference does not seem to be valid, since it is possible that Gottlob thinks that the Morning Star is illuminated by the Sun, but he does not necessarily know this fact about the Evening Star, even if in reality the two planets are the same, which is the case. Russell's solution was that the phrases *the Morning Star* and *the Evening Star* are not proper names, they only have contextual meanings, hence they are not interchangeable due to formal reasons.[10]

In the epsilon language $\mathcal{L}_{\lambda}^{\forall\varepsilon}$, the definite descriptions are proper names, they are manifested as epsilon terms on the object language level, hence the modelling in terms of the epsilon-language fails the Morning Star Test, and it does not explain the puzzle. Fortunately, hitherto, the Frege Puzzle and the semantic status of the expressions like *the Morning star* are not completely solved. If the phrase *the Morning star* is a rigid designator, as it is done in Kripke's proposal, then the Puzzle is solved. Here, temporarily, not having modal context, *rigid* means that the model designates a single individual in one step, and does not select first a set, then a member of it, by a choice function.[11] Then the puzzle only says that, if planet Venus is illuminated by the Sun, then planet Venus is illuminated by the Sun. According to Kripke's approach, the problematic case is the sentence *The Evening Star is the Morning Star*. It is a necessary truth, but it may be problematic from an epistemological point of view.[12] For the epsilon model, the solution is the same. According to Monk, the closed epsilon terms are constants, therefore they are rigid designators in accordance with the Kripke doctrine. However, as Fitting pointed out, an epsilon term, being description-like, can neither be a constant, nor a variable. It is a complex flexible designator (see Fitting 1972). Here, if *the Morning star* is a complex demonstrative (selected by a descriptive term in the actual world), then it is a rigid designator (see Kaplan 1989). Clearly, now, I do not have to deal with the modal context of epsilon terms, knowing that the highly applicable tool of demonstratives might make the modal approach much more complex, and might not add essentially more to the above consideration.

[10] See Russell (1905) and Whitehead & Russell (1910/1967).

[11] Of course, it is a rough simplification. Picking an individual means direct reference, rigid means the term has the same semantic value along the possible worlds. What is more, the notion of "rigid" above is understandable, but mathematically vague.

[12] See Kripke (1972, 102). The whole story can be found in Zvolenszky (2007).

4.2. The King of France Test

Consider the following two sentences

> "The present King of France is bald."
> "The present King of France is not bald."

In order to determine the truth value of the first one, let us imagine the set of all bald people. Since the present King of France is not in this set, the first sentence is false. But, the same reasoning leads to the fact that the second sentence is false too. Which is a contradiction. Hence, the phrase *the present King of France* is not a proper name, it cannot have a meaning in isolation, rather it only has a contextual meaning and the sentences containing such phrases are quantified formulas. This is Russell's solution. In the epsilon calculus the semantic values of the epsilon terms are defined in all cases. The two sentences above are unproblematic, they assign to the phrase *the present King of France* an existing individual as reference. And it is either bald or not bald. According to Theorem 1 of the present paper, sentences *may* possess contextual meaning too, where the truth value is also well-defined. Of course, the reference of *the present King of France* in the epsilon calculus is not the present King of France. Approaching the situation on a more formal level, let us consider the symbolic sentence

$$(\varepsilon x)(x \neq x) = (\varepsilon x)(x \neq x)$$

This is a sentence containing terms which are ill-defined as descriptions: $x \neq x$ is an empty predicate. However, the semantic value of $(\varepsilon x)(x \neq x)$, in a given model, is well-defined. Moreover, $(\varepsilon x)(x \neq x) = (\varepsilon x)(x \neq x)$ is an epsilon invariant sentence, since, it is true in any given epsilon semantics. And indeed, there are epsilon semantics (for example the Bourbaki group's formal systems), where $(\varepsilon x)(x \neq x) = (\varepsilon x)(x \neq x)$ is syntactically identical to the sentence $(\forall x)(x = x)$. $(\varepsilon x)(x \neq x) = (\varepsilon x)(x \neq x)$ is an epsilon-invariant sentence, which has contextual meaning too: it is equivalent to the fact that every individual is identical to itself.

The situation is very similar to the problem of the interesting-looking man holding a martini. In this case, the *the present King of France* is rather a person who is, in fact, bald, but not the present King of France, and $(\varepsilon x)(x \neq x)$ is an existing individual, which is identical to itself, but of course, it does not hold that it is not the same as itself.

Acknowledgements

I would like to thank Endre Latabár, András Simon and an anonymous reviewer for their helpful comments concerning specific parts of the paper or the whole one.

References

Ahrendt, W. and M. Giese. 1999. Hilbert's epsilon-terms in automated theorem proving. In N. V. Murray (ed.) Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX '99. 171–185.

Blass, A. and Y. Gurevich. 2000. The logic of choice. The Journal of Symbolic Logic 65. 1264–1310.

Caicedo, X. 1995. Hilbert's ε symbol in the presence of generalized quantifiers. In M. Krynicki, M. Mostowski and L. W. Szczerba (eds.) Quantifiers: Logics, models and computation. Volume two: Contributions. Berlin & New York: Springer. 63–78.

Donnellan, K. 1966. Reference and definite descriptions. Philosophical Review 75. 281–304.

Dummett, M. 1973. Frege: Philosophy of language. London: Duckworth.

Fitting, M. 1972. An epsilon-calculus system for first order S4. In W. Hodges (ed.) Conference in Mathematical Logic – London '70. Berlin & New York: Springer. 103–110.

Frege, G. 1892/1990. On sense and reference. In P. Geach and M. Black (eds.) Translations from the philosophical writings of Gottlob Frege. Oxford: Basil Blackwell. 56–78.

Kaplan, D. 1989. Demonstratives. In J. Almog, J. Perry and H. Wettstein (eds.) Themes from Kaplan. Oxford: Oxford University Press. 481–563.

Kneebone, G. T. 1963. Mathematical logic and the foundation of mathematics. London: Van Nostrand.

Kripke, S. A. 1972. Naming and necessity. In D. Davidson and G. Harman (eds.) Semantics of natural language. Dordrecht: Reidel. 251–355.

Mints, G. 1996. Thoralf Skolem and the epsilon substitution method for predicate logic. Nordic Journal of Philosophical Logic 1. 133–146.

Molnár, Z. 2013. Epsilon-invariant substitutions and indefinite descriptions. Logic Journal of the IGPL 21. 812–829.

Monk, J. D. 1976. Mathematical logic. Berlin & New York: Springer.

Russell, B. 1905. On denoting. Mind 14. 479–493. (In: B. Russell: Logic and knowledge. London: Routledge. 39–56).

Simmons, H. 2000. Derivation and computation: Taking the Curry–Howard correspondence seriously. Cambridge: Cambridge University Press.

Slater, H. 2007. Completing Russell's logic. Journal of Bertrand Russell Studies 27. Article 15.

Slater, H. 2009. Hilbert's epsilon calculus and its successors. In D. M. Gabbay and J. Woods (eds.) Handbook of the history of logic. Logic from Russell to Church, Vol. 5. Amsterdam: North-Holland. 385–448.

Sørensen, M. H. B. and P. Urzyczyn. 1998. Lectures on the Curry–Howard Isomorphism. Ms. http://disi.unitn.it/∼bernardi/RSISE11/Papers/curry-howard.pdf.

Tarski, A. 1956. Logic semantics, metamathematics papers from 1923 to 1938. Oxford: Clarendon Press.

Troelstra, A. S. and H. Schwichtenberg. 2000. Basic proof theory (Second edition). Cambridge: Cambridge University Press.

Whitehead, A. N. and B. Russell. 1910/1967. Incomplete symbols: Descriptions. In J. van Heijenoort (ed.) From Frege to Gödel: A source book in mathematical logic 1879–1931. Cambridge, MA: Harvard University Press. 216–223.

Zvolenszky, Zs. 2007. Modality, names and descriptions. Doctoral dissertation. New York University.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 445–459

Intuition and decidability in grammar and number theory

GEOFFREY K. PULLUM
University of Edinburgh
Geoffrey.Pullum@ed.ac.uk

KEYWORDS

grammar
prepositions
decidability
algorithm
arithmetic

ABSTRACT

Could a grammatical English sentence contain three consecutive strictly transitive prepositions? One might easily think not: strictly transitive prepositions require NP complements. However, prepositions can be stranded, clausal constituents can begin with prepositions, and so on. Ideally one would like such questions to be algorithmically decidable. I examine the theoretical issue, note a parallel in number theory, reveal the solution to the empirical puzzle (but not the number-theoretic one), and conclude by noting that there is indeed an algorithm for deciding whether some sequence can appear as a proper subsequence of a grammatical string, provided English is context-free.

1. A puzzle in English grammar

Some English prepositions can be used either with a noun-phrase complement or without it if the context permits:

(1) a. Open the gate and walk through it.
    b. Open the gate and walk through.

(2) a. After a while we went back inside the house.
    b. After a while we went back inside.

Others, like *of* and *into*, strictly require a noun-phrase complement and cannot be used grammatically without one:

(3) a. She asked for dihydrocodeine, but I had never heard of it.
    b. *She asked for dihydrocodeine, but I had never heard of.

(4) a. We went back to the car and got into it.
    b. *We went back to the car and got into.

I'll refer to the latter kind of prepositions as *strictly transitive*. Consider now the question stated in (5).

(5) Is there a grammatical English sentence containing a sequence of three consecutive occurrences of a single strictly transitive preposition?

You might think that this question can be immediately answered in the negative on the grounds that the next word after a strictly transitive preposition would have to be the beginning of a noun phrase and noun phrases do not begin with prepositions. But not so fast: English syntax is much trickier than that. For one thing, the complement of a preposition does not have to immediately follow it, but can be displaced in at least three ways.

First, there are prepositional passives like (6b), where what is understood as the complement of a preposition (as in (6a)) is in grammatical terms the subject of the clause, and thus separated from the preposition:

(6) a. People seldom speak of this.
b. This is seldom spoken of.

Grammarians speak of such prepositions as *stranded*. Because of what are often called "subject raising" constructions, the subject can appear arbitrarily far away from the stranded preposition:

(7) This seems to have turned out under the circumstances to have been only very seldom spoken of.

Second, items like interrogative or relative words or phrases can be displaced an arbitrary distance to the beginning of a clause. This is the most common way in which a preposition can be stranded. Notice that these two sentences are both grammatical, and are synonymous (though while the first is normal in style, the second is rather formal and pompous):

(8) a. Which regulation do you think the committee imagines the provost's action might have been in violation of?
b. Of which regulation do you think the committee imagines the provost's action might have been in violation?

In both of these the preposition *of* is understood to have *which regulation* as its complement. Notice that in (8b) the construction involved allows the strictly transitive preposition *of* to be the first word in its clause.

Third, phrases can also be displaced toward the end of the clause, yielding a different way in which a strictly transitive preposition may fail to be immediately followed by its complement, as when a parenthetical is inserted after a preposition:

(9) It was a painting of, or perhaps I should say a painting apparently intended to vaguely suggest, a cornfield in summer.

Thus in order to settle the question in (5) it will be necessary to ensure that no facts of this sort can interact to create a way in which three transitive prepositions could become adjacent. It is not just a matter of which words are allowed to be adjacent to which other words: the interactions of the many different syntactic constructions in English are not necessarily going to be easy to foresee.

2. Generative grammars and decidability

What (5) asks is whether some combination of grammatical configurations can permit a sentence to contain a sequence like *of of of* or *into into into*. It seems intuitively unlikely. But can we prove that it is impossible?

There are systematic computational ways of answering some kinds of questions about sentences in languages. The great majority of the relevant work has been based on systems of rules that Post (1943) originally called production systems, and computer scientists often call rewriting systems, and linguists call generative grammars. Basically they are sets of rules for nondeterministic random construction of abstract structures such as strings or trees.

Post's systems were developed for the purpose of formalizing rules of inference in logic, and were very elaborate, allowing for conclusions to be derived from arbitrary-sized finite sets of premises of arbitrary complexity. As soon as the concept of recursively enumerable (r.e.) sets was clearly formulated in the 1930s, it was clear to Post that any r.e. set of strings could be generated by one of his "canonical production systems". In 1943 he proved that this remained true for dramatically restricted systems that he called "normal" systems, in which every rule had the form "$yX \Rightarrow Xz$" for specified strings y and z, where X is a free string-valued variable, and in 1947 he proved the same for another special case, where the rules all have the form "$X_1yX_2 \Rightarrow X_1zX_2$", for specified strings y and z and variables X_1 and X_2. This, of course, is exactly the form of the grammars that Chomsky (1959) later called "type 0".

For any form of grammar that has this kind of expressive power (i.e., that can generate any arbitrary r.e. set), questions of the form "Does grammar G generate any string containing the substring w?" are always going to be undecidable. This follows from Rice's theorem (Hopcroft & Ullman 1979, 185–192) as applied to generative grammars rather than Turing machines. All non-trivial properties of r.e. sets (that is, properties that hold of some r.e. sets but not all) are undecidable.

I suspect it will also hold for the restricted class called *context-sensitive* languages (equivalent to the class of type 0 grammars in which the z is always at least as long as the y), though the conjecture needs a proof. The rationale for my conjecture is that nearly all decision problems asking for a property of an arbitrary context-sensitive stringset L are undecidable, the two exceptions being membership ("Is w in L?"), which is decidable, and complement type ("Is $\overline{L}$ context-sensitive?"), which was proved in 1987, surprisingly, to be trivial in the sense of having a positive answer for every context-sensitive L (this follows from the Immerman–Szelpcsényi theorem; see Immerman 1999, 149–151). Despite the decidability of membership, context-sensitive languages are extremely similar to arbitrary c.e. sets, and have essentially all of their complexity. Any type 0 grammar G over a symbol inventory Σ can be converted into a context-sensitive grammar G' over $\Sigma \cup b$ where b is a new dummy symbol used for padding the ends of rules in which the right hand side is shorter than the left hand side. $L(G)$ is then obtainable from $L(G')$ simply by ignoring b (where "ignoring b" means applying a homomorphism that erases b).

The most interesting family of stringsets for purposes of studying the properties of human languages is the much smaller subset known as the *context-free* stringsets (standardly called CFLs). This deserves closer attention. CFLs are generated by context-free grammars (CFGs). It is by no means implausible that the set of grammatical sentences of English could be exactly generated by a CFG: see Pullum & Gazdar (1982), Pullum (1985), and Pullum & Rawlins (2007) for discussion of some failed counterarguments.

CFGs are far more tractable than context-sensitive grammars in most respects. But even for a CFG, it is not always immediately obvious what it can do. As a very simple example, consider the CFG with terminals $\{a, b\}$, nonterminals $\{S, A, B\}$, start symbol S, and the rules shown in (10).

(10)
$$\begin{array}{ll} S \to aB & A \to bAA \\ S \to bA & B \to b \\ A \to a & B \to bS \\ A \to aS & B \to aBB \end{array}$$

The strings that this grammar generates are jumbles of *a*'s and *b*'s in arbitrary orders, but they all meet a special condition: the number of *a*'s is exactly the same as the number of *b*'s. The grammar in (10) generates all and only the strings meeting that condition. This could hardly be said to be immediately evident from looking at the rules, but it can be proved inductively (see Hopcroft & Ullman 1979, 81–82).

It follows that the grammar definitely allows for the construction of a sentence with *aaa* in it, and that any such string will also contain at least three instances of *b*, and so on. Indeed, we know for any arbitrary string of *a*'s and *b*'s that the answer to whether (10) can generate it is yes. However, that is specific to (10), and depends on the proof concerning what stringset it generates. Can we decide such questions more generally?

After all, although membership ("Can the string w be generated by the CFG G?") is decidable, and so is emptiness ("Are any strings at all generated by the CFG G?"), many other semantic questions about CFGs (i.e., questions not about their form but about their meaning, in the sense of what strings they can generate under their usual interpretation) are formally undecidable. These include *intersection emptiness* ("Does CFL L_1 have a non-empty intersection with CFL L_2?"), *stringset inclusion* ("Are all the sentences of CFL L_1 included among the sentences of CFL L_2?"), *regularity* ("Is CFL L accepted by a finite-state automaton?"), (see Hopcroft & Ullman 1979, 281), and many others.

The set of all strings of English words (whether grammatical or not) in which the substring *of of of* appears is clearly regular (finite-state), assuming only that English has a finite vocabulary V of words.[1] The finite automaton accepting the set remains always in its start state q_0, checking only that each word is in V, and always rejects, except that if it encounters an *of* it switches to q_1, and if another one immediately follows that it goes into q_2, and if another immediately follows that it goes into q_3. Once in q_3 it always accepts provided only that all subsequent words are in V. We seek a general algorithm for finding out whether some specific CFL has a non-null intersection with that regular set. But the general question of whether two stringsets have a non-null intersection is undecidable.

That is not in contradiction with what was said above about the rules in (10) and the set of strings containing *aaa*. There we had a specific CFG

[1] Kornai (2002) gives an interesting argument against this assumption, based on empirical facts about statistical properties of text: English text exhibits properties that are best modelled in terms of an infinite word stock. But for the sake of the present argument we continue under the usual formal language theory assumption of a finite terminal vocabulary.

for which it happened to be possible to construct a proof that all the generated strings had equinumerous *a*s and *b*s, and that some generated strings contained *aaa*. This shows that in certain special cases we may find out the answer. That does not give us a general algorithm for all cases.

I will return later to the question of whether, given a complete generative grammar for English, there would be a systematic general way of using it to guarantee answers to questions like (5). But first I want to note an interesting similarity to a question in mathematics.

3. A parallel in number theory

Question (5) has a particular logical property in common with the question in (11), which derives from a famous conjecture in number theory.

(11) Is there an even number that is not equal to the sum of two primes?

This can be easily stated using first-order logic interpreted in the usual number-theory model where the domain is the non-negative integers with the operations "+" (addition) and "·" (multiplication). The predicate "even" can be defined as in (12a); "prime" can be defined as in (12b); and then (11) is the question of whether (12c) is true in the specified model.

(12) a. $\mathbf{even}(x) =_{\mathrm{df}} (\exists y[y \geq 1 \wedge y \cdot 2 = x])$

b. $\mathbf{prime}(x) =_{\mathrm{df}} (\neg(\exists y \exists z[y \geq 2 \wedge z \geq 2 \wedge y \cdot z = x]))$

c. $\exists x[\mathbf{even}(x) \wedge \neg(\exists y \exists z[\mathbf{prime}(y) \wedge \mathbf{prime}(z) \wedge y + z = x])]$

What (11) is in effect asking for is a counterexample to the strong Goldbach conjecture, henceforth GC, which claims that every even number greater than 2 is the sum of a pair of primes. Most number theorists are inclined to think this conjecture is true. One reason is that as we consider larger and larger even integers n, the number of different pairs of primes that sum to n increases, so that for any large n it is overwhelmingly likely that there is at least one pair that sums to n. But GC is a non-probabilistic claim, and as is well known, no proof of it has been found, so currently it cannot be guaranteed that the answer to (11) is negative.

The logically interesting property that (5) and (11) share is that for each of them, showing that the question is algorithmically undecidable would ipso facto (though indirectly) tell us the answer. This sounds paradoxical, but it is not. It is a fairly simple point, fairly well known among number theorists and mathematical logicians.

Consider (11) first. To say that the answer to (11) cannot be discovered by an algorithm would mean that GC is unprovable within our system for proving things in arithmetic. And we know from Gödel (1931), of course, that some truths of arithmetic are unprovable in any system capable of expressing all arithmetical truths.

For concreteness, assume the standard system defined by the Peano axioms, known as PA, and a monadic second-order logic interpreted on $\langle \mathbb{N}, +, \cdot \rangle$ (the natural numbers with addition and multiplication). To say that GC is incapable of proof within PA is in effect to say that GC is *independent* of PA, in the sense that we could add GC to the set of PA's consequences, or add its negation, without losing consistency either way. You could believe all the truths of PA plus GC, or believe all the truths of PA plus the negation of GC, and no one would be able to use PA to prove you wrong either way.

Yet if GC were shown to be independent of PA, we would immediately know whether it was true or not: it would have to be true, so the answer to (11) would be negative. Here is the reasoning.

If GC were false, there would be a counterexample, an even number that cannot be expressed as the sum of any two primes. Let g be that number. We could demonstrate GC's falsity in an elementary way by simply exhibiting the list of all triples $\langle p_1, p_2, k \rangle$ such that p_1 and p_2 are primes and $k \leq g$ and $p_1 + p_2 = k$. The list might be very long, if g were very large, but it would be finite, and could be constructed by a very straightforward computer program. The absence from the list of any case where $k = g$ would falsify GC, and thus answer (11) in the affirmative.

If (11) cannot be answered in the affirmative by a proof, the answer to it must be negative, i.e., GC must be true. The answer to (11) cannot be positive yet unprovably so.

An analogous result holds for (5). If we found some way, using facts about a generative grammar for English, to show that (5) cannot be answered within some system of mathematical reasoning, then we would immediately (but indirectly) know that the answer to (5) is negative, because otherwise there would exist a sentence containing three consecutive occurrences of a single transitive preposition, and simply exhibiting the derivation of that sentence would settle the question, offering a proof of the positive answer. The answer to (5) cannot be positive yet unprovably so.

4. The answer to the grammatical puzzle

A key difference between question (5) and question (11) is that (5) is in a sense empirical. It is an empirical fact that those who describe themselves as speakers of English invariably regard *All cows eat grass* as grammatical but **Grass eat cows all* as ungrammatical; they regard *Never have I heard such nonsense* as grammatical but **Never I have heard such nonsense* as ungrammatical; and so on. If there is a sentence containing three instances of some transitive preposition in a row that English speakers treat as grammatical when it is presented to them, then that is an empirical fact (subject to all the usual epistemological caveats, to be discussed very shortly).

My guess, on the basis of 40 years' experience of working on English syntax and techniques for formalizing syntactic theories, and six years working with Rodney Huddleston on the largest and most complete reference grammar currently available for English (Huddleston & Pullum 2002), would have been that the answer to (5) was negative: I would have thought that the rules of English grammar could not allow three consecutive occurrences of a strictly transitive preposition, on the grounds that there wouldn't appear to be any context in which all three of them could have the obligatory noun-phrase complements they require.

But it is a very important fact about argument and evidence in syntax that the intuition of a grammarian regarding generalizations of this sort cannot be trusted.

It is true that the intuition of a native speaker (whether a grammarian or not) can generally be trusted on individual sentences. This is why determining grammatical well-formedness for a sentence of reasonable length normally involves little more than having a native speaker look at it or listen to it, provided some minimal conditions of attentiveness are respected. But caveats are necessary even to that claim, because aspects of meaning, style, phonology, or processing may interfere with intuitive judgments about sentencehood. For example, (13a) will generally be judged grammatical, but the synonymous (13b) will not.

(13) a. Everybody whom everybody left departed.
b. Everybody everybody left left.

This apparently because center-embedding a phrase or clause inside another, even once increases the processing load substantially. (Notice that the relative clause everybody left is embedded with parts of the main clause either side of it, which means processing of the main clause must be interrupted by the processing of another clause and then resumed where it left off; this is discussed in Miller & Chomsky (1963) and much subsequent psycholinguistic literature.) The combination of that with two pairs of adjacent duplicate words is confusing enough to completely wreck the chances of recognizing the grammatical structure.

Likewise, it is well known that there are sentences that confuse us into thinking they are ungrammatical by (as it were) tempting us to process them incorrectly. They are known as *garden-path sentences* (Bever 1970). One celebrated example, well known from the psycholinguistic literature, is (14):

(14) The horse raced past the barn fell.

Our tendency to process this with *raced* as the preterite-tense verb of the main clause, and an unneeded extra verb *fell* on the end, is almost irresistible, and blinds us to the fact that *raced* can also be a past participle, so *raced past the barn* could be a nonfinite passive clause modifying *horse*. In other words, the sentence can be read with the same structure as (15):

(15) The car driven past the barn crashed.

Many other similar examples could be given of the ways in which poor acceptability may wrongly make a properly-formed sentence seem ungrammatical.

However, even if native speakers can in typical cases intuitively perceive the well-formed structure of an individual sentence, even skilled syntacticians cannot reliably intuit the truth of generalizations about wide ranges of sentences or phrases.

The young Noam Chomsky ventured in a conference discussion the assertion that "The verb *perform* cannot be used with mass-word objects: one can perform *a task*, but one cannot perform *labor*" (Hill 1962, 29). Challenged by another participant (Anna Granville Hatcher) to say how he knew this, he answered: "Because I am a native speaker of the English Language". Later in the discussion Hatcher asked him what he would say if the non-count noun were *magic*, and Chomsky was immediately forced to confess: "I think I would have to say that my generalization was wrong" (*ibid.*, 31).

On the specific point of whether a short sentence like *They can perform magic* is grammatical, or whether a short string of words like **They can perform justice* is ungrammatical, he could supply reliable intuition reports, like most native speakers; but confirming a broader generalization about English sentence structure is a very different matter.

And to return to the case at hand, judging whether three consecutive transitive prepositions is possible in English is a judgment concerning an indefinitely large range of sentences. I would have hazarded the guess that the answer was negative, but I would have been wrong. The answer to question (5) is now known, thanks to Wells Hansen (personal communication), and it is positive. Hansen showed this by constructing and exhibiting, rather surprisingly, a grammatical sentence with an *at at at* sequence. A similar one is given in (16).

(16) Donald Trump was laughed at at at least three dinner parties in Manhattan this year.

It is fully grammatical (as well as probably also true), and surprisingly simple to understand. Of course, it might be impugned for style: a writer who notices that some word has been used three times within a short space, or that a jingle effect has been created by two or three words with a similar sound will generally reword. But that is about stylistic acceptability, not grammaticality.

Retrospectively, we can see why and how the Hansen sentence has to be regarded as grammatical. *At* occurs as a grammaticized preposition syntactically required in the construction *laugh at*:

(17) a. They laughed at him.
b. *They laughed to him.
c. *They laughed by him
d. *They laughed on him.

And the choice of preposition is determined by the choice of verb; other verbs require different prepositions:

(18) a. *We spoke at him. (*speak* does not take *at*)
b. We spoke to him. (*speak* does take *to*)
c. *We spoke on him. (*speak* does not take *on*)

(19) a. *They rely at him. (*rely* does not take *at*)
b. *They rely to him. (*rely* does not take *to*)
c. They rely on him. (*rely* does take *on*)

Verb-preposition combinations of this sort readily yield prepositional passives (*was laughed at*, *was spoken to*, *was relied on*, etc.). Hansen's sentence has the form of a prepositional passive clause, with the first *at* of the sequence as its stranded preposition. The subject of the clause (*Donald Trump*) is understood as the complement of the first *at*.

But *at* is somewhat more syntactically versatile than *of* in one respect: it can also serve as the head of a locative modifier like *at three parties*, which can occur following a prepositional passive; and it occurs in the idiomatic Preposition + Adjective combination *at least*, which can serve as an adjunct modifying a determinative like *three*, hence, crucially, can stand at the beginning of a noun phrase, as in *at least three parties*, and thus can begin a noun phrase serving as the complement of the preposition *at*.

Thus when the first *at* is stranded in a prepositional passive construction it is possible for a second *at*-phrase heading a locative adjunct to follow, and for a third *at*-phrase to begin the noun phrase within that locative adjunct. All those facts are relevant to why it is that *at at at* can be a possible subsequence in a grammatical sentence.

5. Proper substring possibility for CFLs

We should never forget that English syntax constitutes a vast domain of exploration, within which are many known unknowns, and an unknown number of unknown unknowns. This domain cannot be explored via the simplistic appeals to "logic" that purists and usage advisers so often advocate. Which sentences are grammatical is not determined by any kind of common-sense or formal logic. The grammatical sentences are simply the ones that happen to be permissible under the set of rules or constraints that defines the language – the large set of exception-ridden and often rather quirky rules that define English as it happens to be today. Discovering how we are to precisely formulate the content of those rules is a major scientific enterprise. Even an informal survey of the ground that must be covered takes up more than 1,700 pages of text (Huddleston & Pullum 2002, henceforth *CGEL*).

And we cannot blithely assume, even when we have produced such a grammar, that there will exist an algorithm for finding out whether it is possible for a grammatical sentence to meet some given condition, such as having three consecutive transitive prepositions, or containing the sequence *and the of*, or any other syntactically definable property of symbol strings. Indefinitely many precisely framed questions about human languages (considered as stringsets over a word vocabulary) are undecidable, even given a full, exact, and correct grammar for the language (which even for English, of course, we do not have as yet).

While in general native speakers (whether grammarians or not) have intuitive reactions concerning the grammaticality of specific strings of words presented to them, they do not have intuitional access to the truth values of generalizations about the entire range of sentences that are grammatical in their language, any more than mathematicians have intuitional access to the truth values of generalizations about the integers. The key difference is that we take the truths of number theory to be a priori and necessary, substantiable through rigorous proof as in the other formal sciences, while the true statements about English grammar are at root empirical.

We have qualified intuitional access to the status of specific sentences because we subconsciously respond to them as if we were encountering them in actual use, and to some extent we can report on our responses (see Devitt 2006, chapter 7, for a discussion of this topic that I find very perceptive). But we have no veridical intuitional access to broader generalizations about the grammar of our language, and can be surprised by discovering them.

Questions about whether some word sequence like *at at at* or *of of of* can form a subsequence of a grammatical sentence in some human language, if we assume for concreteness that human languages can be generated by CFGs, have the general form seen in (20):

(20) Proper substring possibility

Given a CFG G with terminal vocabulary V and an arbitrary string w in V^*, does G generate any string that has w as a proper substring?

One might ask whether there could ever be practical reasons for needing answers to such questions. Practical importance is of course something that in general we discover only retrospectively, but it is not impossible to imagine a context in which information about occurrence of substrings might be of practical use to an engineer. A robot equipped to parse and understand spoken English might be designed with a special simple word sequence that would immobilize it to permit servicing (or to block a *Westworld*-style disaster in which robots become malign). To make sure the robot could not be immobilized unintentionally through ordinary conversation, one might want the word sequence to be one that definitely could not form part of any sentence of the language. We know, thanks to Wells Hansen's discovery, that *at at at* could not serve that purpose. Maybe *of of of* could.

So is (20) formally decidable? The answer turns out to be yes. A problem closely related to it was studied by Lang (1988) in the context of devising a context-free parsing algorithm that will yield useful output even when faced with sentences containing unknown parts of unknown length, by producing a finite representation of the set of all possible parses (perhaps infinitely many) that could allow for the missing parts. Subsequently Osorio and Navarro (2001) tackled more directly the problem of solving proper substring possibility as stated in (20), using the CKY algorithm as the basis for their proof and showing that the problem can be decided in cubic time.

Osorio and Navarro point out that the problem actually has many more areas of potential application than you might think, since CFG parsing is so closely related to other computational tasks like matrix multiplication and is so widely applicable: it could be relevant to DNA analysis in bioinformatics, syntax-driven development of language tools, and shape analysis in pattern recognition.

Given a CFG for English, therefore, we could use a fully general algorithm to find out (in cubic time) whether, for example, there is a grammatical string with *of of of* as a substring. (I think there probably is, but I leave the exercise of constructing one for the reader to pursue in idle moments.) Of course, the algorithm presupposes the completion of a CFG that fully and accurately generates all and only the sentences of English. The informal account in *CGEL*, mentioned above, is not expressed in anything like the form of a CFG.

For what it is worth, Pullum and Rogers (2008) provide, in a rather unexpected way, good reason to believe that there is nothing in *CGEL* that is beyond the power of CFGs. They note that although the objects that *CGEL* uses as structural representations are not trees, they are very close to being trees, and the very limited departure from treehood that is employed (downward convergence of branches in certain particular kinds of noun phrase) can be described by a transduction to covering trees expressible in weak monadic second-order logic (wMSO), and wMSO-describable sets of trees always have CFLs as their frontier sets (by the theorem of Doner 1970). Hence *CGEL* appears to covertly entail that English (considered as a stringset over a vocabulary of dictionary words) is a CFL.

In principle, then, there almost certainly exists a CFG for English on which we could run an algorithm of the sort sketched by Osorio and Navarro to find out whether *of of of* (or any other word sequence) can be a substring of a sentence.

Acknowledgements

My thanks to Wells Hansen, whose serendipitous observation of an *at at at* sentence inspired this paper; to Colin Stirling, who discovered that the substring admissibility problem had been shown to be decidable by Osorio and Navarro (2001); to Marcus Kracht (originally an anonymous referee), who suspected that this was true and drafted a different proof from Osorio and Navarro's, later generalizing it to cover all multiple context-free grammars (MCFGs); and to my old friend and research collaborator András Kornai, who has taught me so much (though it will never be enough) about mathematically informed work on human languages.

References

Bever, T. G. 1970. The cognitive basis for linguistic structures. In J. R. Hayes (ed.) Cognition and the development of language. New York: Wiley. 279–362.

Chomsky, N. 1959. On certain formal properties of grammars. Information and Control 2. 137–167.

Devitt, M. 2006. Ignorance of language. Oxford: Clarendon.

Doner, J. 1970. Tree acceptors and some of their applications. Journal of Computer and System Sciences 4. 406–451.

Gödel, K. 1931. Über formal unentscheidbare Sätze der *Principia mathematica* und verwandter Systeme I. Monatshefte für Mathematik und Physik 38. 173–198. (English translation in *Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931*, ed. by Jean van Heijenoort, Harvard University Press, Cambridge MA, 1967, 596–616.)

Hill, A. A. 1962. Third Texas Conference on Problems of Linguistic Analysis in English. Austin, TX: University of Texas.

Hopcroft, J. E. and J. D. Ullman. 1979. Introduction to automata theory, languages and computation. Reading, MA: Addison-Wesley.

Huddleston, R. and G. K. Pullum. 2002. The Cambridge grammar of the English language. Cambridge: Cambridge University Press.

Immerman, N. 1999. Descriptive complexity. New York: Springer.

Kornai, A. 2002. How many words are there? Glottometrics 4. 61–86.

Lang, B. 1988. Parsing incomplete sentences. In Proceedings of the 12th Conference on Computational Linguistics, Volume 1, COLING '88. Stroudsburg, PA: Association for Computational Linguistics, 365–371.

Miller, G. A. and N. Chomsky. 1963. Finitary models of language users. In R. D. Luce, R. R. Bush and E. Galanter (eds.) Handbook of mathematical psychology, vol. 2. New York: Wiley. 419–491.

Osorio, M. and J. A. Navarro. 2001. Decision problem of substrings in context free languages. In CIC-X: Memorias del X Congreso Internacional de Computación. 239–249.

Post, E. L. 1943. Formal reductions of the general combinatory decision problem. American Journal of Mathematics 65. 197–215.

Post, E. L. 1947. Recursive unsolvability of a problem of Thue. Journal of Symbolic Logic 12. 1–11.

Pullum, G. K. 1985. On two recent attempts to show that English is not a CFL. Computational Linguistics 10. 182–186.

Pullum, G. K. and G. Gazdar. 1982. Natural languages and context-free languages. Linguistics and Philosophy 4. 471–504.

Pullum, G. K. and K. Rawlins. 2007. Argument or no argument? Linguistics and Philosophy 30. 277–287.

Pullum, G. K. and J. Rogers. 2008. Expressive power of the syntactic theory implicit in *The Cambridge grammar of the English language*. Paper presented at the 49th Annual Meeting of the Linguistics Association of Great Britain, University of Essex.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 461–479

An infrastructure for empowering internet users to handle fake news and other online media phenomena

GEORG REHM
DFKI GmbH, Language Technology Lab
georg.rehm@dfki.de

KEYWORDS

fake news
online misinformation
annotation
language technology
computational linguistics

ABSTRACT

Online media, online news and online communication have an unprecedented and increasing level of social, political and also economic relevance. This article proposes an infrastructure to address phenomena of modern online media production, circulation and reception by establishing a distributed architecture that relies on automatic processing and human feedback.

1. Introduction

Usually lumped together under the "fake news" label, a bundle of novel topics around online media production, circulation, reception and their impact has emerged in recent years and, thus, is receiving a lot of attention from multiple stakeholders including politicians, journalists, researchers, industry, non-governmental organisations (NGOs) and civil society. In addition to the challenge of addressing and dealing with "fake news", "alternative facts" as well as "post-truth politics", there is an ever increasing amount of hate speech, abusive language and cyber bullying taking place online.[1]

Among the interested stakeholders are politicians who have begun to realise that, increasingly, major parts of public debates and social discourse are carried out online, on a small number of social networks. We have witnessed that not only online discussions but also the perception of trends, ideas, theories, political parties, individual politicians, elections and societal challenges can be subtly influenced and significantly rigged using targeted social media campaigns, devised at manipulating opinions to create long-term sustainable mindsets on the side of the recipients. We live in

[1] A revised version of this paper was published in Rehm & Declerck (2017).

a time in which online media, online news and online communication have an unprecedented level of social, political and also economic relevance.

Due to the sheer importance and visibility of the topic one cannot help but think about designing and deploying technologies to improve the situation, maybe even to solve the problem altogether – thanks to the recent breakthroughs in artificial intelligence (AI) (Metz 2016; Gershgorn 2016; Martinez-Alvarez 2017; Chan 2017) –, while at the same time *not* putting in place a centralised infrastructure that could be misused for the purpose of censorship, media manipulation or mass surveillance.[2]

This paper addresses key challenges of the digital age (section 2) by introducing and proposing a technological framework concept (section 3), which has been devised under the umbrella of a two-year research and technology transfer project, in which a research centre collaborates with four small and medium-sized enterprise (SME) partners that face the challenge of having to process, analyse and make sense of large amounts of digital content. The companies cover four different use cases and sectors (Rehm & Sasaki 2015) including journalism. For these partners we develop a platform that provides access to language and knowledge technologies Bourgonje et al. (2016a;b). The services are integrated by the SME partners into their own in-house systems or those of clients (Rehm et al. 2017). Among others, we currently develop services aimed at the detection and classification of abusive language and clickbait content.

2. Online media in 2017: status quo

The debate around online media is currently dominated by several topics and challenges. They share certain characteristics that make it possible to address them with the same technological approach.

A key prerequisite for the current situation is the existence of the World Wide Web itself: everybody is able to create content, to write an article on a certain topic. Until a few years ago the key challenge was to optimise the HTML code, linking and metadata to get into the top of the relevant search engine results pages for important keywords. Nowadays, however, content is no longer predominantly discovered through search engines but through social media platforms: users see interesting content, which is then shared to their own connections. Many users only read a

[2] An indicator for the relevance of the topic is the increasing number of "how to identify fake news" articles published online (Mantzarlis 2015; Bazzaz 2016; Rogers & Bromwich 2016; Wardle 2017; Walbrühl 2017).

headline, identify a certain relevance to their own life and then spread the content. When in doubt, users estimate the trustworthiness of the source: potentially dubious stories about which they are skeptical are shared anyway if the source or friend through whom the story was discovered is considered reliable or if the number of views is rather high, which, to many users, indicates legitimacy.

There is a tendency for very provocative, aggressive, one-sided, allegedly "authentic" (Marchi 2012) content. The idea is to make it as easy as possible to identify the stance of the article so that the reader's own world view is validated, implicitly urging the user to share the content. The hope of the content's originator is that a story will go viral, that it will be shared very quickly by many users and spread through multiple networks in order to establish a reach of millions of people. One sub-category of this type of content is "clickbait", articles with dubious factual content that are presented with misleading headlines, designed for the purpose of generating many clicks. The more extreme the virality, the higher the reach, the higher the click numbers, the higher the advertisement revenue. The term "clickbait" is usually associated with commercial intentions, but it can also refer to articles spreading political mis- or disinformation.

Content is, first and foremost, discovered through a small number of big social networks. While only a handful of search engines and online news outlets used to be the central points of information until a few years ago, the role of the centralised hub is now played by social networks that help content to be discovered and go viral (Barthel et al. 2016). All social networks have the same key feature, a feed or timeline, i.e., posts, news, tweets, photos that are presented to the user, starting at the most recent one. As there is simply too much content, all social networks introduced machine learning-based algorithms to determine which content to present to a certain user. They are continuously trained through interactions with the network, i.e., "liking" a post boosts the respective topic, checking the feed of a certain friend on the network boosts the connection to this friend. Some networks have even introduced more fine-grained sentiments that can be used in addition to the simple "like" (see, e.g., Facebook's reactions "love", "haha", "wow", "sad", "angry"). Through "likes" of topics, connections to friends and interactions with the site, the social network creates, and continuously updates, for every single user, an internal model of likes and interests. This model is used to select content to be presented on the timeline by only selecting content that is assessed as being relevant to the user's interests. Plus, algorithms typically favour content that is being "liked" or shared by those friends and connections that the user interacts with the most. This is the origin of the filter bubble phenomenon: users are only exposed to content that can also be described as “safe” – content shared by friends they know and like is considered content that matches a user’s interests. Controversial content that contradicts a user’s world view or that presents opposing information, that challenges their beliefs is *not* presented – according to the underlying user model it is not relevant.

In the digital age, we can no longer assume that everything that has been published is necessarily correct. While this has been true in some parts of the world for decades, this challenge has now also entered the Western part of the world. Since November 2016 it has been socially accepted, in some parts of the political spectrum, to categorise fact-checked articles, written by experienced journalists and published by respected news outlets, as “fake news” – not because the news are false but because the corresponding articles do not endorse and support the opinion and agenda of the reader. The age of post-factual politics creates an unprecedented tension and stimulates fundamental debates about the relationship between politics and the fourth estate in civil society and beyond.

Additionally, we are faced with the challenge that more and more content is produced and spread with the sole purpose of manipulating the readers’ beliefs and opinions by appealing to their emotions instead of informing them objectively. Rather, this type of opinionated, emotional, biased, often aggressive and far-right content is prepared and spread to reach specific goals, for example, to create support for controversial ideas or to destroy the reputation of a politician. These coordinated online marketing campaigns are often carried out by experts with in-depth knowledge of the underlying technologies and processes. They involve large numbers of bots and fake accounts as amplifiers Weedon et al. (2017) as well as large budgets for online advertisements in social media, clearly targeted at very specific demographic groups the originators want to influence and then to flip to reach a specific statistical threshold. The way news are nowadays spread, circulated, read and shared – with less and less critical thinking or fact checking – enables this type of content to gather a large number of readers (and sharers) quickly. The filter bubble acts like an echo chamber that can amplify any type of content, from genuine, factual news to emotionally charged, politically biased news, to false news to orchestrated disinformation campaigns, created with the specific purpose of large-scale manipulation. Content of the last two categories can be hard or very hard to identify even for human experts.

A key challenge is to separate objective, balanced content, be it journalistic or user-generated, from hateful, abusive or biased content, maybe produced with a hidden agenda. Even if fundamentally different in nature, nowadays both types of content share the same level of visibility, reach and exposure through the equalisation mechanisms of the social web, which can be easily manipulated. In the past the tasks of fact checking, critical thinking and unveiling hidden agendas have mostly been in the realm of journalism, but in the digital age they are more and more transferred to the actual reader and recipient of online content. The analysis, curation and assessment of content is no longer carried out by professional journalists or news editors – the burden of fact checking and content verification is left to the reader. This aspect is getting even more crucial because the number of people who state that social networks are their *only* source of news and information is growing steadily (Marchi 2012). The most prominent example from recent history is that social media manipulation can apparently even make or break a national election (Barthel et al. 2016; Rogers & Bromwich 2016; Marwick & Lewis 2017). It must be noted, though, that a large number of fact checking initiatives is active all over the world (Mantzarlis 2017), but they mostly rely on human expertise and, thus, do not scale (Martinez-Alvarez 2017; Dale 2017). The small number of automated fact checking initiatives are fragmented (Babakar & Moy 2016).

Several types of online content are often grouped together under the label "fake news". For example, Holan (2016) defines fake news as "made-up stuff, masterfully manipulated to look like credible journalistic reports that are easily spread online to large audiences willing to believe the fictions and spread the word." In reality, the situation is much more complex. Initially based on the classification suggested by Wardle (2017), Table 1 (overleaf) shows a first attempt at bringing together the different types of false news including selected characteristics and associated intentions. The table shows the complexity of the situation and that a more fine-grained terminology is needed to discuss the topic properly, especially when it comes to designing technological solutions that are meant to address one or more of these types of content.

An additional challenge is the proliferation of hateful comments and abusive language, often used in the comments and feedback sections on social media posts. The effects can be devastating for the affected individual. Many hateful comments on repeated postings by the same person, say, a pupil, are akin to cyberbullying and cybermobbing. There is also a clear tendency to aggressive comments on, for example, the social media pages of traditional news outlets, who have to ask the users more and more to behave in a civilised way.

Table 1: Characteristics and intentions associated with different types of false news (adapted from Wardle 2017; Walbrühl 2017; Rubin et al. 2015; Holan 2016; Weedon et al. 2017)

	Satire or parody	False connection	Misleading content	False context	Imposter content	Manipulated content	Fabricated content
Clickbait		X	X	?		?	?
Disinformation			X	X		X	X
Politically biased		?	X	?		?	X
Poor journalism		X	X	X			
To parody	X				?		X
To provoke					X	X	X
To profit	?	X			X		X
To deceive		X	X	X	X	X	X
To influence politics			X	X		X	X
To influence opinions			X	X	X	X	X

3. Technology framework: approach

Technically, online content is predominantly consumed through two possible channels, both of which rely substantially on World Wide Web technology and established web standards. Users either read and interact with content directly on the web (mobile or desktop versions of websites) or through dedicated mobile apps; this can be considered using the web implicitly as many apps make heavy use of HTML5 and other web technologies. The World Wide Web itself still is and, for the foreseeable future, will continue to be the main transport medium for online content. The suggested technology architecture is, hence, designed as an additional layer on top of the web. Nevertheless, we also have to be clear about the scope and ambition of the challenge: the infrastructure needs to be able to cope with millions of users, arbitrary content types, hundreds of languages and massive amounts of data. The goal is to empower and to enable users to balance out the network, echo chamber and filter bubble effects and to provide mechanisms to filter for abusive content.

3.1. Services of the infrastructure

In many cases the burden of analysing and fact checking online content has been shifted to the reader (section 2), which is why corresponding analysis and curation services need to be made available in an efficient and ubiquitous way. The same tools to be used by *content consumers* can and should also be applied by *content creators*, e.g., journalists and bloggers. Those readers who are interested to know more about what they are currently reading should be able to get the additional information as easily as possible, and the same applies to those journalists who are interested in fact-checking the content they are researching for the production of new content.

Readers of online content are users of the World Wide Web. They need, first and foremost, web-based tools and services with which they can process any type of content to get additional information on a specific piece, be it one small comment on a page, the main content component of a page (for example, an article) or even a set of interconnected pages (one article spread over multiple pages), for which an assessment is sought.

The provided services need to be designed to operate in and with the web stack of technologies, i.e., within the web ecosystem, they need to support users in their task of reading and curating content within the browser in a smarter and, eventually, more balanced way. This can be accomplished by providing additional, also alternative opinions and view points, by presenting other, indepedent assessments, or by indicating if content is dangerous, abusive, factual or problematic in any way. Fully automatic technologies (Rubin et al. 2015; Schmidt & Wiegand 2017; Horne & Adal 2017; Martinez-Alvarez 2017) can take over a subset of these responsibilities but, given the current state of the art, not all, which is why the approach needs to be based both on simple or complex automatic filters and watchdogs as well as human intelligence and feedback.[3]

The tools and services should be available to every web user without the need to install any additional third-party software. This is why these services, ideally, should be integrated into the browser on the same level as bookmarks, the URL field or the navigation bar, i.e., without relying on the installation of a plugin. The curation tools should be thought of as an inherent technology component of the World Wide Web, for which intuitive and globally acknowledged user-interface conventions can be established,

[3] A fully automatic solution would work only for a very limited set of cases. A purely human-based solution would work but required large amounts of experts and, hence, would not scale. This is why we favour, for now, a hybrid solution.

such as, for example, traffic light indicators for false news content (green: no issues found; yellow: medium issues found and referenced; red: very likely false news). Table 2 shows a first list of tools and services that could be embedded into such a system.[4] Some of these can be conceptualised and implemented as automatic tools (Horne & Adal 2017), while others need a hybrid approach that involves crowd-sourced data and opinions. In addition to displaying the output of these services, the browser interface needs to be able to gather, from the user, comments, feedback, opinions and sentiments on the current piece of content, further to feed the crowd-sourced data set. The user-generated data includes both user-generated annotations (UGA) and also user-generated metadata (UGM). Automatically generated metadata are considered machine-generated metadata (MGM).

Table 2: Suggested tools and services to be provided through the infrastructure (selection)

Tool or Service	Description	Approach
Political bias indicator	Indicates the political bias (Martinez-Alvarez 2017) of a piece of content, e.g., from far left to far right	automatic
Hate speech indicator	Indicates the level of hate speech a certain piece of content contains	automatic
Reputation indicator	Indicates the reputation, credibility (Martinez-Alvarez 2017), trustworthiness, quality (Filloux 2017) of a certain news outlet or individual author of content	crowd, automatic
Fact checker	Checks if claims are backed up by references, evidence, established scientific results and links claims to the respective evidence (Babakar & Moy 2016)	automatic
Fake news indicator	Indicates if a piece of content contains non-factual statements or dubious claims (Horne & Adal 2017; Martinez-Alvarez 2017)	crowd, automatic
Opinion inspector	Inspect opinions and sentiments that other users have with regard to this content (or topic) – not just the users commenting on one specific site, but all of them	crowd, automatic

[4] This list is meant to be indicative rather than complete. For example, services for getting background information on images are not included (Gupta et al. 2013). Such tools could help pointing out image manipulations or that an old image was used, out of context, to illustrate a new piece of news.

3.2. Characteristics of the infrastructure

In order for these tools and services to work effectively, efficiently and reliably, they need to possess several key characteristics, which are quintessential for the overall success of the approach.

Like the Internet and the World Wide Web, the infrastructure must be operated in a federated, i.e., de-centralised setup – a centralised approach would be too vulnerable for attacks or misuse. Multiple organisations, companies, research centres or NGOs should be able to set up, operate and offer services (section 3.1) and additional pieces of the infrastructure. The internal design of the respective algorithms and tools may differ substantially, but their output (MGM) should comply to a standardised metadata format. It is rather likely that political biases in different models meant to serve the same purpose cannot be avoided, which is especially likely for models based on large amounts of data, which, in turn, may inherently include a political bias. This is why users must be enabled to activate or deactivate as many of these tools as they want to get an aggregated value, for example, with regard to the level of hate speech in content or its political bias. Services and tools must be combinable, i.e., they need to comply to standardised input and output formats (Babakar & Moy 2016). They also need to be transparent (Martinez-Alvarez 2017). Only transparent, i.e., fully documented, checked, ideally also audited approaches can be trustworthy.

Access to the infrastructure should be universal and available everywhere, i.e., in any browser, which essentially means that, ideally, the infrastructure should be embedded into the technical architecture of the World Wide Web. As a consequence, access mechanisms should be available in every browser, on every platform, as native elements of the graphical user interface (GUI). These functions should be designed in such a way that they support users without distracting them from the content. Only if these tools are available virtually anywhere, can the required scale be reached.

The user should be able to configure and to combine multiple services, operated in a de-centralised way, for a clearly defined purpose in order to get an aggregated value. There is a danger that this approach could result in a replication and shift of the filter bubble effect (section 2) onto a different level but users would at least be empowered actively to configure their own personal set of filters to escape from any resulting bubble. The same transparency criterion also applies to the algorithm that aggregates multiple values, of course.

3.3. Building blocks of the infrastructure

Research in Language Technology and Natural Language Processing (NLP) currently concentrates on smaller components, especially watchdogs, filters and classifiers (see section 4) that could be applied under the umbrella of a larger architecture to tackle current online media phenomena (section 2). While this research is both important and crucial, even if fragmented and somewhat constrained by the respective training data sets (Rubin et al. 2015; Conroy et al. 2015; Schmidt & Wiegand 2017) and limited use cases, we also need to come to a shared understanding how these components can be deployed and made available. The suggestion consists of the following building blocks (see Figure 1).

3.3.1. Building block: natively embedded into the World Wide Web

An approach that is able to address modern online media and communication phenomena adequately needs to operate on a web-scale level. It should natively support cross-lingual processing and be technically and conceptually embedded into the architecture of the World Wide Web itself. It should be standardised, endorsed and supported not only by all browser vendors but also by all content and media providers, especially the big social networks and content hubs. Only if *all* users have *immediate* access to the tools and services suggested in this proposal can they reach its full potential. The services must be unobtrusive and cooperative, possess intuitive usability, their recommendations and warnings must be immediately understandable, and it must be simple to provide general feedback (UGM) and assessments on specific pieces of content (UGA).

3.3.2. Building block: web annotations

Several pieces of the proposed infrastructure are already in place. One key component are Web Annotations, standardised by the World Wide Web Consortium (W3C) in early 2017 (Sanderson 2017; Sanderson et al. 2017a;b). They enable users to annotate arbitrary pieces of web content, essentially creating an additional and independent layer on top of the regular web. Already now Web Annotations are used for multiple individual projects in research, education, scholarly publishing, administration and investigative journalism.[5] Web Annotations are *the* natural mechanism to

[5] See, for example, the projects presented at I Annotate 2015 (http://iannotate.org/2015/), 2016 (http://iannotate.org/2016/) and 2017 (http://iannotate.org/2017/).

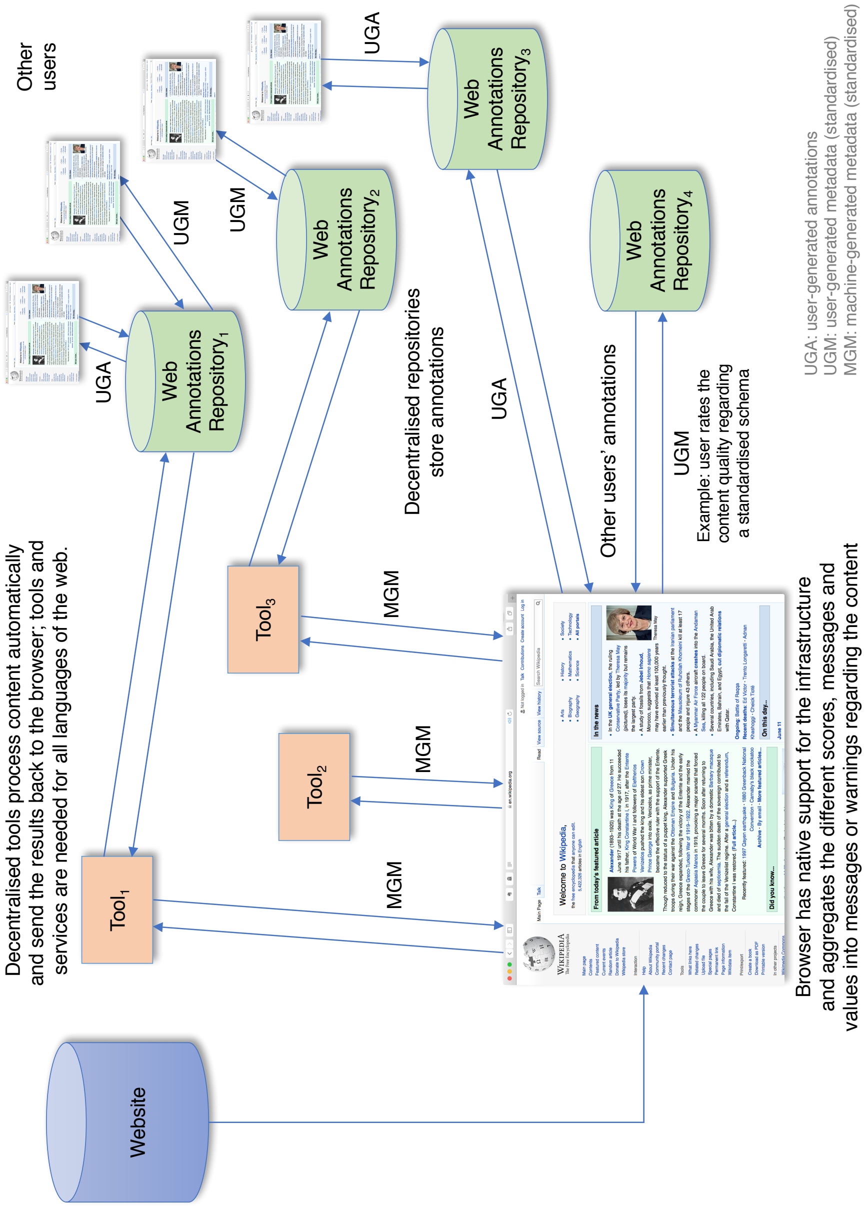


Figure 1: Simplified architecture of the proposed infrastructure

enable users and readers interactively to work with content, to include feedback and assessments, to ask the author or their peers for references or to provide criticism. The natural language content of Web Annotations (UGA) can be automatically mined using methods such as sentiment analysis or opinion mining – in order to accomplish this across multiple languages, this needs to be done cross-lingually (Rehm et al. 2016). However, there are still limitations. Content providers need to enable Web Annotations by referencing a corresponding JavaScript library. Federated sets of annotation stores or repositories are not yet foreseen, neither are native controls in the browser that provide aggregated feedback, based on automatic (MGM) or manual content assessments (UGM, UGA). Another barrier for the widespread use and adoption of Web Annotations are proprietary commenting systems, as used by all major social networks. Nevertheless, services such as Hypothes.is enable Web Annotations on any web page, but native browser support, ideally across all platforms, is still lacking. A corresponding browser feature needs to enable both free-text annotations of arbitrary content pieces (UGA), but also very simple flagging of problematic content, for example, "content pretends to be factual but is of dubious quality" (UGM). Multiple UGA, UGM or MGM annotations could be aggregated and presented to new readers of the content to provide guidance and indicate any issues.

3.3.3. Building block: metadata standards

Another needed piece of the architecture is an agreed upon metadata schema Babakar & Moy (2016) to be used both in manual annotation scenarios (UGM) and also by automatic tools (MGM). Its complexity should be as little as possible so that key characteristics of a piece of content can be adequately captured and described either by humans or machines. With regard to this requirement, W3C published several standards to represent the provenance of digital objects (Groth & Moreau 2013; Belhajjame et al. 2013a). These can be thought of as descriptions of the entities or activities involved in producing or delivering a piece of content to understand how data was collected, to determine ownership and rights or to make judgements about information to determine whether to trust content (Belhajjame et al. 2013b). An alternative approach is for content publishers to use Schema.org's ClaimReview[6] markup in their websites after specific facts have been checked. The needed metadata schema can be based on the W3C provenance ontology and/or Schema.org. Additional metadata fields are likely to be needed.

[6] https://schema.org/ClaimReview

3.3.4. Building block: tools and services

Web Annotations can be used by readers of online content to provide comments or to include the results of researched facts (UGA, UGM). Automatic tools and services that act as filters and watchdogs can make use of the same mechanisms (MGM, see section 3.1). These could be functionally limited classifiers, for example, regarding abusive language, or sophisticated natural language understanding (NLU) components that attempt to check certain statements against one or more knowledge bases. Regardless of the complexity and approach, the results can be made available as globally accessible Web Annotations (that can even, in turn, be annotated themselves). Services and tools need to operate in a decentralised way, i.e., users must be able to choose from a wide variety of automatic helpers. These could, for example, support users to position content on the political spectrum, either based on crowd-sourced annotations, automatic tools, or both.

3.3.5. Building block: decentralised repositories and tools

The setup of the infrastructure must be federated and decentralised to prevent abuse by political or industrial forces. Data, especially annotations, must be stored in decentral repositories, from which browsers retrieve, through secure connections, data to be aggregated and displayed (UGM, UGA, MGM, i. e., annotations, opinions, automatic processing results etc.). In the medium to long term, in addition to annotations, repositories will also include more complex data, information and knowledge that tools and services will make use of, for example, for fact checking. In parallel to the initiative introduced in this article, crowd-sourced knowledge graphs such as Wikidata or DBpedia will continue to grow. The same is true for semantic databases such as BabelNet and many other data sets, usually available and linkable as Linked Open Data. Already now we can foresee more sophisticated methods of validating and fact-checking arbitrary pieces of content using systems that make heavy use of knowledge graphs, for example, through automatic entity recognition and linking, relation extraction, event extraction and mapping etc. One of the key knowledge bases missing, in that regard, is a Web Annotation-friendly event-centric knowledge graph, against which fact-checking algorithms can operate.[7] Basing algorithms that are supposed to determine the truth of a statement on automatically extracted and formally represented knowledge creates both

[7] GDELT (Global Database of Events, Language, and Tone) comes close but is lacking with regard to its integratability, see http://www.gdeltproject.org.

practical and philosophical questions, among others, who checks these automatically extracted knowledge structures for correctness? How do we represent conflicting view points and how do algorithms handle conflicting view points when determining the validity of a statement? How do we keep the balance between multiple subjective opinions and an objective and scientific ground-truth?

3.3.6. Building block: aggregation of manual and automatic annotations

The final key building block of the proposed system relates to the aggregation of manual and automatic annotations, created in a de-centralised and highly distributed way by human users and automatic services (UGA, UGM, MGM). Already now we can foresee very large numbers of annotations so that the aggregation and consolidation will be a non-trivial challenge. This is also true for those human annotations that are not based on shared metadata vocabularies but that are free text – for these free and flexible annotations, robust and also multilingual annotation mining methods need to be developed.

4. Related work

Research on Computer-Mediated Communication (CMC) has a long tradition. Scholars initially concentrated on different types of novel communication media such as e-mail, IRC, Usenet newsgroups, and different hypertext systems and document types, especially personal home pages, guestbooks and, later, discussion fora. Early on, researchers focused upon the (obvious) differences between these new forms of digital communication and the traditional forms, especially when it comes to linguistic phenomena that can be observed on the text surface (smileys, emoticons, acronyms etc.). Several authors pointed out that the different forms of CMC have a certain oral and spoken style, quality and conceptualisation to them, as if produced spontaneously in a casual conversation, while being realised in a written medium (Haase et al. 1997).

If we now fast forward to 2017, a vastly different picture emerges. About half of the global population has access to the internet, most of whom also use the World Wide Web and big social networks. The internet is no longer considered fringe technology that is only used by scientists, early adopters and computer nerds, but it is mainstream. Nowadays the internet acts like an amplifier and enabler of social trends. It continues to penetrate and to disrupt our lives and social structures, especially our established traditions of social and political debates. The relevance of online media, online news and online communication could not be any more crucial. While early analyses of CMC, e.g., Reid (1991), observed that the participants were involved in the "deconstruction of boundaries" and the "construction of social communities", today the exact opposite seems to be case: not only online but also offline can we observe the (disturbing) trend of increased, intricately orchestrated, social and political manipulation, nationalism and the exclusion of foreigners, immigrants and seemingly arbitrary minorities – boundaries are constructed, social communities deconstructed, people are manipulated, individuals excluded.

There is a vast body of research on the processing of online content including text analytics (sentiment analysis, opinion and argument mining), information access (summarisation, machine translation) and document filtering (spam classification). Attempting to classify, among others, the different types of false news shown in Table 1 requires, as several researchers also emphasise, a multi-faceted approach that includes multiple different processing steps. We have to be aware of the ambition, though, as some of the "fake news detection" use case scenarios are better described as "propaganda detection", "disinformation detection", maybe also "satire detection". These are difficult tasks at which even humans often fail. Current research in this area is fragmented and concentrates on very specific sub-problems, see, for example, the Fake News Challenge, the Abusive Language Workshop, or the Clickbait Challenge.[8] What is missing, however, is a practical umbrella that pulls the different pieces together and that provides an approach that can be realistically implemented and deployed including automatic tools as well as human annotations.

5. Summary and conclusions

Humanity is transitioning into becoming a digital society, or at least a "digital first" society, i.e., news, media, facts, rumours (Zubiaga et al. 2016; Srivastava et al. 2017), information are created, circulated and disseminated online. Already now the right social media strategy can make or break an election or is able to influence if a smaller or larger societal or demographic group (city, region, country, continent) is in favour or against constructively solving a certain societal challenge. Social media and online communication can be extremely powerful tools to bridge barriers, to in-

[8] See http://www.fakenewschallenge.org, http://www.clickbait-challenge.org, https://sites.google.com/site/abusivelanguageworkshop2017/.

form people and to enable global communication. When abused, misused or infiltrated, they are a dangerous weapon.

The fields of Computational Linguistics, Language Technology and Artificial Intelligence should actively contribute solutions to this key challenge of the digital age. If we don't, there is a concrete danger that stakeholders with bad intentions are able to influence parts of the society to their liking, only constrained by their political, commercial, egotistical interests. Technologies need to be developed to enable every user of online media to break out of their filter bubbles and to inform themselves in a balanced way, taking all view points into account.

After dumb digital content, smart content and semantic content enrichment we now need to concentrate on content curation tools that enable *contextualised content*, i.e., content that can be, ideally, automatically cross-referenced and fact-checked, and for which additional background information can be retrieved in a robust way. This can involve assessing the validity of claims and statements made in the content as well as retrieving related texts, facts and statements, both in favour and against a certain piece of content.

Next steps include presenting this proposal in various different fora and communities, among others, researchers and technologists, standards-developing organisations (Babakar & Moy 2016) and national as well as international political bodies. At the same time, research needs to be continued and prototypes of the architecture as well as individual services developed, enabling organisations to build and to deploy decentralised tools early. While a universal, globally accessible, balanced and well maintained knowledge graph containing up-to-date information about entities and events would be handy to have, it is out of scope with regard to the initiative reported here; it is safe to assume that such a knowledge repository will be developed in parallel in the next couple of years. The proposed architecture can be used to link online content against this knowledge graph and to measure the directions of online debates.

The proposal introduced in this article is ambitious in its scope and implications, prevention of misuse will play a hugely important role. How can we make sure that a certain piece of technology is only used with good intentions? Recently it has been shown that a user's social media data can reliably predict if the user is suffering from alcohol or drug abuse (Ding et al. 2017). Will this technology be used to help people or to stigmatise them? Will an infrastructure, as briefly sketched in this paper, be used to empower users to make up their own minds by providing additional information about online content or will it be used to spy on them and to manipulate them with commercial or political intentions?

Acknowledgements

The project "Digitale Kuratierungstechnologien" (DKT) is supported by the German Federal Ministry of Education and Research (BMBF), "Unternehmen Region", instrument Wachstumskern-Potenzial (no. 03WKP45). More information: http://www.digitale-kuratierung.de.

References

Babakar, M. and W. Moy. 2016. The state of automated factchecking – How to make factchecking dramatically more effective with technology we have now. https://tinyurl.com/yd6cdoo5

Barthel, M., A. Mitchell and J. Holcomb. 2016. Many Americans believe fake news is sowing confusion. http://www.journalism.org/2016/12/15/many-americans-believe-fake-news-is-sowing-confusion/

Bazzaz, D. 2016. News you can use: Infographic walks you through 10 questions to detect fake news. http://www.seattletimes.com/education-lab/infographic-walks-students-through-10-questions-to-help-them-spot-fake-news/

Belhajjame, K., J. Cheney, D. Corsark, D. Garijo, S. Soiland-Reyes, S. Zednik and J. Zhao. 2013a. PROV-O: The PROV Ontology. Tech. rep., World Wide Web Consortium (W3C). https://www.w3.org/TR/2013/REC-prov-o-20130430/

Belhajjame, K., H. Deus, D. Garijo, G. Klyne, P. Missier, S. Soiland-Reyes and S. Zednik. 2013b. PROV Model Primer. W3C Working Group Note, World Wide Web Consortium (W3C). https://www.w3.org/TR/2013/NOTE-prov-primer-20130430/

Bourgonje, P., J. Moreno-Schneider, J. Nehring, G. Rehm, F. Sasaki and A. Srivastava. 2016a. Towards a platform for curation technologies: Enriching text collections with a semantic-web layer. In H. Sack, G. Rizzo, N. Steinmetz, D. Mladeni, S. Auer and C. Lange (eds.) The semantic web. Berlin & New York: Springer, no. 9989 in Lecture Notes in Computer Science. 65–68.

Bourgonje, P., J. M. Schneider, G. Rehm and F. Sasaki. 2016b. Processing document collections to automatically extract linked data: Semantic storytelling technologies for smart curation workflows. In A. Gangemi and C. Gardent (eds.) Proceedings of the 2nd International Workshop on Natural Language Generation and the Semantic Web (WebNLG 2016). Edinburgh: The Association for Computational Linguistics, 13–16.

Chan, R. 2017. Artificial intelligence is going to destroy fake news – But A.I. can also cause the volume of fake news to explode. https://www.inverse.com/article/27723-artificial-intelligence-will-destroy-fake-news

Conroy, N. J., V. L. Rubin and Y. Che. 2015. Automatic deception detection: Methods for finding fake news. Proceedings of the Association for Information Science and Technology 52. 1–4.

Dale, R. 2017. NLP in a post-truth world. Natural Language Engineering 23. 319–324.

Ding, T., W. K. Bickel and S. Pan. 2017. Social media-based substance use prediction. https://arxiv.org/abs/1705.05633

Filloux, F. 2017. Quality for news is mostly about solving the reputation issue. https://mondaynote.com/quality-for-news-is-mostly-about-solving-the-reputation-issue-fdebd0dcc9e2

Gershgorn, D. 2016. In the fight against fake news, artificial intelligence is waging a battle it cannot win. https://qz.com/843110/can-artificial-intelligence-solve-facebooks-fake-news-problem/

Groth, P. and L. Moreau. 2013. PROV-Overview: An overview of the PROV family of documents. Tech. rep., World Wide Web Consortium (W3C). https://www.w3.org/TR/prov-overview/

Gupta, A., H. Lamba, P. Kumaraguru and A. Joshi. 2013. Faking Sandy: Characterizing and identifying fake images on Twitter during Hurricane Sandy. In Proceedings of the 22nd International Conference on World Wide Web. Rio de Janeiro, 729–736.

Haase, M., M. Huber, A. Krumeich and G. Rehm. 1997. Internetkommunikation und Sprachwandel. In R. Weingarten (ed.) Sprachwandel durch Computer. Opladen: Westdeutscher Verlag. 51–85.

Holan, A. D. 2016. 2016 Lie of the year: Fake news. http://www.politifact.com/truth-o-meter/article/2016/dec/13/2016-lie-year-fake-news/

Horne, B. D. and S. Adal. 2017. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In Proceedings of the 2nd International Workshop on News and Public Opinion at ICWSM.

Mantzarlis, A. 2015. 6 tips to debunk fake news stories by yourself. http://www.poynter.org/2015/6-tips-to-debunk-fake-news-stories-by-yourself/385625/

Mantzarlis, A. 2017. There are now 114 fact-checking initiatives in 47 countries. https://www.poynter.org/2017/there-are-now-114-fact-checking-initiatives-in-47-countries/450477/

Marchi, R. 2012. With Facebook, blogs, and fake news, teens reject journalistic "objectivity". Journal of Communication Inquiry 36. 246–262.

Martinez-Alvarez, M. 2017. How can Machine Learning and AI help solving the fake news problem? https://miguelmalvarez.com/2017/03/23/how-can-machine-learning-and-ai-help-solving-the-fake-news-problem/

Marwick, A. and R. Lewis. 2017. Media manipulation and disinformation online. https://datasociety.net/output/media-manipulation-and-disinfo-online/

Metz, C. 2016. The bittersweet sweepstakes to build an AI that destroys fake news. http://www.wired.com/2016/12/bittersweet-sweepstakes-build-ai-destroys-fake-news/

Rehm, G. and T. Declerck (eds.). 2017. Sprachtechnologien für die Herausforderungen des digitalen Zeitalters – Language Technologies for the Challenges of the Digital Age: Proceedings of the GSCL Conference 2017. Berlin: Gesellschaft für Sprachtechnologie und Computerlinguistik. http://gscl2017.dfki.de

Rehm, G. and F. Sasaki. 2015. Digitale Kuratierungstechnologien – Verfahren für die effiziente Verarbeitung, Erstellung und Verteilung qualitativ hochwertiger Medieninhalte. In Proceedings of the 2015 international conference of the german society for computational linguistics and language technology. GSCL '15, 138–139.

Rehm, G., F. Sasaki and A. Burchardt. 2016. Web Annotations – A game changer for language technologies? Presentation given at I Annotate 2016, Berlin. http://www.slideshare.net/georgrehm/web-annotations-a-game-changer-for-language-technology; http://iannotate.org/2016/

Rehm, G., J. M. Schneider, P. Bourgonje, A. Srivastava, J. Nehring, A. Berger, L. König, S. Räuchle and J. Gerth. 2017. Event detection and semantic storytelling: Generating a travelogue from a large collection of personal letters. In T. Caselli, B. Miller, M. van Erp, P. Vossen, M. Palmer, E. Hovy and T. Mitamura (eds.) Proceedings of the events and stories in the news workshop. Vancouver, Canada: Association for Computational Linguistics.

Reid, E. M. 1991. Electropolis: Communication and community on Internet Relay Chat. Honours Thesis. University of Melbourne. http://www.aluluei.com/electropolis.htm

Rogers, K. and J. E. Bromwich. 2016. The hoaxes, fake news and misinformation we saw on election day. https://www.nytimes.com/2016/11/09/us/politics/debunk-fake-news-election-day.html

Rubin, V. L., Y. Chen and N. J. Conroy. 2015. Deception detection for news: Three types of fakes. Proceedings of the Association for Information Science and Technology 52. 1–4.

Sanderson, R. 2017. Web Annotation Protocol. W3C Recommendation, World Wide Web Consortium (W3C). https://www.w3.org/TR/2017/REC-annotation-protocol-20170223/

Sanderson, R., P. Ciccarese and B. Young. 2017a. Web Annotation Data Model. Tech. rep., World Wide Web Consortium (W3C). https://www.w3.org/TR/2017/REC-annotation-model-20170223/

Sanderson, R., P. Ciccarese and B. Young. 2017b. Web Annotation Vocabulary. Tech. rep., World Wide Web Consortium (W3C). https://www.w3.org/TR/2017/REC-annotation-vocab-20170223/

Schmidt, A. and M. Wiegand. 2017. A survey on hate speech detection using Natural Language Processing. In Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media. Valencia.

Srivastava, A., G. Rehm and J. M. Schneider. 2017. DFKI-DKT at SemEval-2017 Task 8: Rumour detection and classification using cascading heuristics. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Vancouver: Association for Computational Linguistics, 477–481.

Walbrühl, D. 2017. Das musst du wissen, um Fake News zu verstehen. https://perspective-daily.de/article/213/AhopoOEF

Wardle, C. 2017. Fake news. It's complicated. https://firstdraftnews.com/fake-news-complicated/

Weedon, J., W. Nuland and A. Stamos. 2017. Information operations and Facebook. https://fbnewsroomus.files.wordpress.com/2017/04/facebook-and-information-operations-v1.pdf

Zubiaga, A., M. Liakata, R. Procter, G. W. S. Hoi and P. Tolmie. 2016. Analysing how people orient to and spread rumours in social media by looking at conversational threads. PLoS ONE 11. https://doi.org/10.1371/journal.pone.0150989

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 481–497

The definition of Named Entities

ESZTER SIMON
Research Institute for Linguistics,
Hungarian Academy of Sciences
simon.eszter@nytud.mta.hu

KEYWORDS

computational linguistics
named entity recognition
annotation schemes
proper names
compositionality

ABSTRACT

Named Entity Recognition (NER) is one of the most intensively studied tasks of computational linguistics. It has two substeps: first, locating the Named Entities (NEs) in unstructured texts, and second, classifying them into pre-defined categories. A key issue is how to define NEs. This issue interconnects with the issue of selection of classes and the annotation schemes applied in the field of NER. The major standard guidelines do not give an exact definition of NEs, but rather list examples and counterexamples. For getting a usable definition of NEs, we investigate the approach taken in the philosophy of language and linguistics, and we map our findings to the NER task. We do not wish to give a complete description of the theory and typology of proper names but to find a plausible way to define linguistic units relevant to the NER task.

1. Introduction

Named Entity Recognition (NER), the task of automatic identification of selected types of Named Entities (NEs), is one of the most intensively studied tasks of Information Extraction (IE). Presentations of language analysis typically begin by looking words up in a dictionary and identifying them as nouns, verbs, adjectives, etc. But most texts include lots of names, and if a system cannot find them in the dictionary, it cannot identify them, making it hard to produce a linguistic analysis of the text. Thus, NER is of key importance in many Natural Language Processing (NLP) tasks, such as Information Retrieval (IR) or Machine Translation (MT).

The NER task, which is often called Named Entity Recognition and Classification in the literature, has two substeps: first, locating the NEs in unstructured texts, and second, classifying them into pre-defined categories. A key issue is how to define NEs. This issue interconnects with the issue of selection of classes and the annotation schemes applied in the field of NER.

The major standard guidelines applied in the field of NER do not give an exact definition of NEs, but rather list examples and counterexamples. The only common statement they make is that NEs have unique references. For getting a usable definition of NEs, we investigate the approach taken in the philosophy of language and linguistics, and we map our findings onto the NER task. We do not wish to give a complete description of the theory and typology of proper names, but to find a plausible way to define linguistic units relevant for the NER task.

The article is structured as follows.[1] In section 2, we give an overview of the annotation schemes applied in the field of NER. Section 3 describes the philosophical approach, and section 4 gives the linguistic background of the theory of proper names. The article concludes in section 5 with the most important findings about mapping the theory of proper names to the NER task.

2. Annotation schemes

2.1. MUCs

The first major event dedicated to the NER task was the 6th Message Understanding Conference *(MUC-6)* in 1995. As the organizers write in their survey about the history of MUCs (Grishman & Sundheim 1996), these conferences were rather similar to shared tasks, because the submission of participants' results was a prerequisite for participation at the conference. Prior MUCs focused on other IE tasks; MUC-6 was the first including the NER task, which consisted of three subtasks (Sundheim 1995):

– entity names (ENAMEX): organizations, persons, locations;
– temporal expressions (TIMEX): dates, times;
– number expressions (NUMEX): monetary values, percentages.

The annotation guidelines define NEs as "unique identifiers" of entities, and give an enormous list of what to annotate as NEs. However, the best support for annotators is the restriction about what not to annotate: "names that do not identify a single, unique entity".

[1] This article is a slightly modified version of a chapter of the author's PhD dissertation (Simon 2013).

As for the temporal expressions, the guidelines distinguish between absolute and relative time expressions. To be considered absolute, the expression must indicate a specific segment of time, e.g.,

(1) twelve o'clock noon

(2) January 1979

A relative time expression indicates a date relative to the date of the document, or a portion of a temporal unit relative to the given temporal unit, e.g.,

(3) last night

(4) yesterday evening

In MUC-6, only absolute time expressions were to be annotated.

The numeric expressions subsume monetary and percentage values. Modifiers that indicate the approximate value of a number are to be excluded from annotation, e.g.,

(5) about 5%

(6) over $90,000

The modified version of the MUC-6 guidelines was used for the *MUC-7* NER task in 1998 (Chinchor 1998). The most notable change was that relative time expressions became taggable. The MUC-7 guidelines became one of the most widely used standards in the field of NER. They were used with slight modifications for the Multilingual Entity Tasks (MET-1 and 2) (Merchant et al. 1996) and for the Hub-4 Broadcast News Evaluation (Miller et al. 1999) in 1999.

According to the MUC guidelines, embedded NEs can also be annotated, e.g.,

(7) The [morning after the [July 17]$_{\text{DATE}}$ disaster]$_{\text{TIME}}$

2.2. CoNLL

The Computational Natural Language Learning *(CoNLL)* conference is the yearly meeting of the Special Interest Group on Natural Language Learning (SIGNLL) of the Association for Computational Linguistics (ACL). Shared tasks organized in 2002 and 2003 were concerned with language-independent NER (Tjong Kim Sang 2002; Tjong Kim Sang & De Meulder 2003). Annotation guidelines were based on the NER task definition of the MITRE Corporation (http://www.mitre.org/) and the Science Applications International Corporation (SAIC) (Chinchor et al. 1999), which are slightly modified versions of the MUC guidelines. A new type, `Measure`, was introduced for NUMEX elements, e.g.,

(8) 23 degrees Celsius

In contrast to the MUC guidelines, instructions are given regarding certain kinds of metonymic proper names, decomposable and non-decomposable names, and miscellaneous non-taggables. The latter constitute a new category, `Miscellaneous`, which includes names falling outside the classic ENAMEX, e.g., compounds that are made up of locations, organizations, etc., adjectives and other words derived from a NE, religions, political ideologies, nationalities, or languages.

2.3. ACE

As part of the *Automatic Content Extraction* (ACE) program (a series of IE technology evaluations from 1999 organized by the National Institute of Standards and Technology (NIST)), new NE types were introduced in addition to the classic ENAMEX categories: `Facility`, `Geo-Political Entity`, `Vehicle` and `Weapon`. The category `Facility` subsumes artifacts falling under the domains of architecture and civil engineering. `Geo-Political Entities` are composite entities comprised of a population, a government, a physical location, and a nation (or province, state, county, city, etc.). The seven main types are divided into dozens of subtypes and hundreds of classes (ACE 2008) . The ACE program is concerned with automatic extraction of content, including not only NEs but also their relationships to each other and events concerning them. For the purposes of this more complex task, all references to entities are annotated: names, common nouns, noun phrases, and pronouns. In this regard, ACE is exceptional in the race of NER standards, where common nouns and pronouns are not to be annotated.

2.4. LDC

The Linguistic Data Consortium (LDC) has developed annotation guidelines for NEs and time expressions within the *Less Commonly Taught Languages* (LCTL) project. In contrast to the ones mentioned above, these guidelines give an exact definition of NEs (LDC 2006) : "An entity is some object in the world – for instance, a place or a person. A named entity is a phrase that uniquely refers to that object by its proper name, acronym, nickname or abbreviation." Besides the classical name categories (`PER, ORG, LOC`), they also annotate `Titles`, which are separated from the person's name, e.g.,

(9) said [GlobalCorp]$_{\text{ORG}}$ [Vice President]$_{\text{TTL}}$ [John Smith]$_{\text{PER}}$

The LCTL annotation guidelines are the first concerned with meaning and compositionality of NEs: "The meaning of the parts of names are not typically part of the meaning of the name (i.e., names are not *compositional*) and, therefore, names cannot be broken down into smaller parts for annotation." Thus, a NE is treated as an indivisible syntactic unit that cannot be interrupted by an outside element.

In addition to the classical ENAMEX, TIMEX and NUMEX categories, there are a wide range of other, marginal types of NEs, which are relevant for particular tasks, e.g., extracting chemical and drug names from chemistry articles (Krallinger et al. 2015); names of proteins, species, and genes from biology articles (Ding et al. 2015); or project names, email addresses and phone numbers from websites (Zhu et al. 2005).

2.5. Summary

Early works define the NER problem as the recognition of proper names in general. Names of persons, locations and organizations have been studied the most. Besides these classical categories, there is a general agreement in the NER community about the inclusion of temporal expressions and some numerical expressions, such as amounts of money and other types of units. The main categories can be divided into fine-grained subtypes and classes, and marginal types are sometimes included for specific tasks. Annotation guidelines usually do not go further in defining NEs than saying that they are "unique identifiers" or that they "uniquely refer" to an entity. Only one of the guidelines mentions the meaning and compositionality of NEs: it postulates NEs as indivisible units, although earlier guidelines allow embedded NEs.

3. Language philosophical views: from Mill to Kripke

3.1. John Stuart Mill

"A proper name is a word that answers the purpose of showing what thing it is that we are talking about, but not of telling anything about it", writes John Stuart Mill in his 1843 *A system of logic* (Mill 2002). According to him, the semantic contribution of a name is its referent and only its referent. One of his examples illustrating this statement is the name of the town Dartmouth. The town was probably named after its localization, because it lies at the mouth of the river Dart. But if the river had changed its course, so that the town no longer lay at the mouth of the Dart, one could still use the name *Dartmouth* to refer to the same place as before. Thus, it is not part of the meaning of the name *Dartmouth* that the town with this name lies at the mouth of the Dart.

3.2. Gottlob Frege and Bertrand Russell

Gottlob Frege's puzzle of the Morning Star and the Evening Star challenges the Millian conception of names. In his famous work *Über Sinn und Bedeutung* (Frege 2000), he distinguishes between *sense (Sinn)* and *reference (Bedeutung)*. Without the distinction between sense and reference, the following sentences would be equal:

(10) The Morning Star is the Evening Star.

(11) The Morning Star is the Morning Star.

Both names have the same reference (Venus), so they should be interchangeable. However, since the thought expressed by (10) is distinct from the thought expressed by (11), the senses of the two names are different. While (11) seems to be an empty tautology, (10) can be an informative statement, even a scientific discovery. If somebody did not know that the Evening Star is the Morning Star, he/she could think that (11) was true, while (10) was false.

To solve the puzzle, without resorting to a two-tiered semantic theory, Bertrand Russell used the description theory. The *description theory of names* states that each name has the semantic value of some definite description (Cumming 2012). For example, *Aristotle* might have the semantic value of 'the teacher of Alexander the Great'. *The Morning Star* and *the Evening Star* might correspond to different definite descriptions in their semantic value, and would make different semantic contributions to the sentences in which the occur.

Frege and Russell both argue that Mill was wrong: a proper name is a definite description abbreviated or disguised, and such a description gives the sense of the name. According to Frege, a description may be used synonymously with a name, or it may be used to fix its reference.

3.3. Saul Kripke

Saul Kripke concurred only partially with Frege's theory. Description fixes reference, but the name denoting that object is then used to refer to that object, even if referring to counterfactual situations where the object does not have the properties in question, writes Kripke in *Naming and necessity* (Kripke 1981). One of Kripke's examples is Gödel and the proof of incompleteness of arithmetic. If it turned out that Gödel was not the man who proved the incompleteness of arithmetic, Gödel would not be called 'the man who proved the incompleteness of arithmetic', but he would still be called 'Gödel'. Thus, names are not equal to definite descriptions.

Kripke postulates proper names as *rigid designator*s. Something is a rigid designator if it designates the same object in every possible world. The concept of a possible world (or counterfactual situation) is used in modal semantics, where the sentence *Frank might have been a revolutionist* is interpreted as a quantification over possible worlds. Kripke suggests an intuitive test to find out what is a rigid designator. An updated example: *the President of the US in 2017* designates a certain man, Trump; but someone else (e.g., Clinton) may have been the President in 2017, and Trump might not have; so this designator is not rigid. When talking about what would happen to Trump in a certain counterfactual situation, we are talking about what would happen to *him*. So 'Trump' is a rigid designator.

With respect to proper names, reference can be fixed in various ways. In the case of initial baptism it is typically fixed by ostension or description. Otherwise, the reference is usually determined by a chain, passing the name from link to link. In general, the reference depends not just on what we think, but on other people in the community, the history of how knowledge of the name has spread. It is by following a history that one gets to the reference.

Kripke argues that proper names are not the only kinds of rigid designators: species names, such as *tiger*, or mass terms, such as *gold*, certain terms for natural phenomena, such as *heat*, and measurement units, such as *one meter* are further examples. There is a difference between the phrase *one meter* and the phrase *the length of the metre bar at* t_0. The first phrase is meant to designate rigidly a certain length in all possible worlds, which in the actual world happens to be the length of the metre bar at t_0. On the other hand, *the length of the metre bar at* t_0 does not designate anything rigidly.

3.4. Summary

Kripke goes back to the Millian theory of names, and at the same time breaks with Frege's theory, when he writes that proper names do not have sense, only reference. He declares that a proper name is a rigid designator, which designates the same object in every possible world. Through examples he proves that definite descriptions are not synonymous with names, but they can still fix a referent. In the case of proper names, the reference can be fixed in an initial baptism, after which the name spreads in the community by a chain, from link to link. In Kripke's theory, species names, mass terms, natural phenomena and measurement units are also rigid designators.

4. The linguistic approach

Besides the theory of rigid designators, another concept used in the literature to define NEs is that of unique reference. In subsection 4.1, we clarify the meaning of the phrase "unique reference", which seems to be used non-systematically in NER guidelines. Unique reference can act as the separator line between proper names and common nouns. There are however certain *linguistic properties* by which we can make a stronger distinction, as described in subsection 4.2. The main feature distinguishing between them is the issue of compositionality, which is discussed in subsection 4.3. Finally, we sum up our findings about the linguistic background of proper names in subsection 4.4.

4.1. Unique reference

In the MUC guidelines (Chinchor 1998), the definition of what to annotate as NEs is as follows: "proper names, acronyms, and perhaps miscellaneous other unique identifiers", and what not to annotate as NEs: "artifacts, other products, and plural names that do not identify a single, unique entity".

In the LCTL guidelines we find the following definition: "a NE is a phrase that uniquely refers to an object by its proper name, acronym, nickname or abbreviation" (LDC 2006).

Let us take these definitions one by one. In the first case, the phrase "unique identifiers" is coordinated with "proper names" and "acronyms", and "unique" is an attributive adjective modifying the noun "identifiers". Thus, "unique" means here that the identifier is unique, similarly to proper names and acronyms. In the second case, however, it is the entity a linguistic unit refers to that must be unique in order for the unit to qualify as a NE. In the LCTL guidelines, the phrase "uniquely refers" means something similar as in the first case, it is therefore the referring linguistic unit that must be unique, not the entity in the world to which it refers.

Here and in several other places in the literature, the difference between the concepts of referring act and reference seems to be blurred. When trying to determine what is unique, we find that in most grammar books the names and the entities they refer to are not clearly distinguished. However, it does matter whether we are talking about Charlie or about the name *Charlie*. To prevent such an ambiguity, we always indicate the meta-linguistic usage by single quotation marks.

By investigating various definitions of proper names, we can conclude that names refer to a unique entity (e.g., *London*), so names have unique reference (Quirk & Greenbaum 1980), in contrast to common nouns, which refer to a class of entities (e.g., *cities*), or non-unique instances of a certain class (e.g., *city*). However, we can refer to and even identify an entity by means of common nouns. The difference is that proper names, even standing by themselves, always identify entities, while a common noun can do so only in such cases when it constitutes a noun phrase with other linguistic units. Common nouns may stand with a possessive determiner (e.g., *my car*), or with a demonstrative (e.g., *this car*), or can be a part of a description (e.g., *the car that I saw yesterday*).

Many proper names share the feature of having only one possible reference, but a wide range of them refer to more than one object in the world. For example, *Washington* can refer to thousands of people who have *Washington* as their surname or given name, a US state, the capital of the US, cities and other places throughout America and the UK, roads, lakes, mountains, educational organizations, and so forth. These kinds of proper names are referentially multivalent (Anderson 2007), but each of the references is still unique.

Some proper names occur in plural form, optionally or exclusively. In the latter case, the plural suffix is an inherent part of the name. These are the so called *pluralia tantum* (e.g., *Carpathians, Pleiades*). According to their surface form, it might seem that they can be broken down into smaller pieces, but the Carpathians do not consist of $carpathian_1$, $carpathian_2$, ..., $carpathian_n$, just as the Pleiades do not consist of pleiades. These names refer to groups of entities considered unique.

Names of brands, artifacts, and other products can be optionally used in plural form. For example, *Volvo* is a proper name referring to a unique company. But if we put it in a sentence, like *He likes Volvos*, it will refer to particular vehicles. This is a kind of metonymy, with the company name used to refer to a product of this company. Proper names in plural form can also be used in other kinds of figures of speech, for example in metaphors. In the phrase *a few would-be Napoleons*, some characteristics of the emperor are associated with men to which the word *Napoleons* refers. In these cases, proper names act like common nouns, i.e., they have no unique reference.

Additionally, there is a quite large number of linguistic units which are on the border between proper names and common nouns, because it is difficult to determine whether their reference is unique. Typically, they are used as proper names in some languages, but as common nouns in other ones. The difficulty of classification is usually mirrorred even in the spelling rules. For example, in the case of events (*World War II, Olympic Games* in English; *2. világháború, olimpiai játékok* in Hungarian; *Segunda Guerra Mundial, Juegos Olímpicos* in Spanish; *Seconde Guerre mondiale, Jeux olympiques* in French), expressions for days of the week and months of the year (*Monday, August* in English; *hétfő, augusztus* in Hungarian; *lunes, agosto* in Spanish; *lundi, août* in French), expressions for languages, nationalities, religions and political ideologies (*Hungarian, Catholic, Marxist* in English; *magyar, katolikus, marxista* in Hungarian; *húngaro, católica, marxista* in Spanish; *hongrois, catholique, marxiste* in French), etc. Categories vary across languages, so there seems to be no language-independent, general rule for classifying proper names.

4.2. Distinction between proper names and common noun phrases

As mentioned above, proper nouns are distinguished from common nouns on the basis of the uniqueness of their reference. However, we can make a stronger distinction based on other linguistic properties.

First, we have to clarify the disctinction between proper nouns and proper names made by current works in linguistics (e.g., Anderson 2007; Huddleston & Pullum 2002). Since the term “noun” is used for a class of single words, only single-word proper names are proper nouns: *Ivan* is both a proper noun and a proper name, but *Ivan the Terrible* is a proper name that is not a proper noun. From this distinction follows that proper names cannot be compared to a single common noun, but to a noun phrase headed by a common noun. A proper noun by itself constitutes a noun phrase, while common nouns need other elements. In subsection 4.1, we gave a few examples. In the subsequent analysis, proper names and common noun phrases are juxtaposed.

Distinction between proper nouns and common nouns is commonly made with reference to *semantic properties*. One of them is the classic approach: entities described by a common noun, e.g., *horse*, are bound together by some resemblances, which can be summed up in the abstract notion of 'horsiness' or 'horsehood' (Gardiner 1957). A proper name, on the contrary, is a distinctive badge: there is no corresponding resemblance among the Charlies that could be summed up as 'Charlieness' or 'Charliehood'. Thus, we can say that common nouns realize abstraction, while proper names make distinction. However, Katz (1972) argues that the meaninglessness of names means that one cannot establish a semantic distinction between proper names and common noun phrases. The latter are compositional, because their meaning is determined by their structure and the meanings of their constituents (Szabó 2008), while proper names "allow no analysis and consequently no interpretation of their elements", quoting Saussure (1959). Thus, proper names are arbitrary linguistic units, and are therefore not compositional (see 4.3 for more details).

Moving on to *syntax*, common noun phrases are compositional, i.e., they can be divided into smaller units, while proper names are indivisible syntactic units. This is confirmed by the fact that proper names – as opposed to common nouns – cannot be modified internally, as can be seen in these examples:

(12) my son's college

(13) my son's beautiful college

(14) beautiful King's College

(15) *King's beautiful College

Further evidence is that in Hungarian and other highly agglutinative languages, the inflection always goes to the end of the proper name constituting a noun phrase. (16) presents the inflection of a proper name (here: a title), while (17) shows its common noun phrase counterpart (consider the second determiner in the latter):

(16) Láttam az Egerek és embereket.
'I saw (Of Mice and Men).ACC'

(17) Láttam az egereket és az embereket.
'I saw the mice.ACC and the men.ACC'

From the perspective of *morphology*, proper names must always be sacred, which means that the original form of a proper name must be reconstructible from the inflected form (Deme 1956). This requirement is mirrorred even in the current spelling rules in Hungarian: e.g., *Papp-pal* 'with Papp', *Hermann-nak* 'to Hermann'. Some proper names in Hungarian have common noun counterparts, as well, e.g., *Fodor* ∼ *fodor* 'frill', *Arany* ∼ *arany* 'gold'. Since the word *fodor* is exceptional, when inflecting it as a common noun, the rule of vowel drop is applied: *fodrot* 'frill.ACC'. However, when inflecting it as a proper name, it is inflected regularly, without dropping the vowel: *Fodort* 'Fodor.ACC'. The common noun *arany* also has exceptional marking, it is lowering, which means that it has *a* as a link vowel in certain inflectional forms, e.g., in the accusative, instead of the regular bare accusative marker: *arany-at* 'gold-ACC'. But as a proper name, it is inflected regularly: *Arany-t* 'Arany-ACC' (for more details, see Kornai 1994 and Kenesei et al. 1998). Psycholinguistic experiments on Hungarian morphology also confirm that proper names are inflected regularly (Lukács 2001), while common nouns may have exceptional markings.

4.3. The non-compositionality of proper names

In order to examine whether proper names are compositional or arbitrary linguistic units, here we give an analysis of how knowledge about the named entity can be deduced from the name. Proper names are not simply arbitrary linguistic units, but they show the arbitrariness most clearly of all, since one can give any name to his/her dog, ship, etc. It follows from the arbitrariness of the initial baptism that proper names say nothing about the properties of the named entity, in fact they do not even indicate what kind of entity we are talking about (a dog, a ship, etc.).

Although *monomorphemic* proper names are classic examples of non-compositionality, they are not semantically empty. For instance, Charlie is a boy by default, but this name is often given to girls in the US, and of course it can be given to pets or products. Semantic implications of proper names (if any) are therefore defeasible. This is in contrast with common nouns, since we cannot call a table 'chair' without violating the

Gricean maxims (Grice 1975). Monomorphemic proper names have only one non-defeasible semantic implication, namely if one is called *X*, then the predicate 'it is called *X*' will be true (cf. the Millian theory of proper names in section 3).

In the context of the current analysis, two types of *polymorphemic* proper names can be distinguished. First, there are phrases which are headed by a common noun and modified by a proper name, e.g., *Roosevelt square, Columbo pub.* The second type consists of two (or more) proper nouns, e.g., *Theodore Roosevelt, Volvo S70.*

In the case of the former, more frequent type, every non-defeasible semantic implication (except the fact of the naming) comes from the head, the modifier does not make any contribution. This can be shown by removing the head: from the sentence *You are called from the Roosevelt*, one cannot determine the source of the call, which might come from the Roosevelt Hotel, from the Roosevelt College, or from a bar in Roosevelt square. All we have is the trivial implication, that Roosevelt is the name of the place. The fact that the modifier contributes nothing to the semantics of the entire construction can be illustrated better by replacing the proper names with empty elements, e.g., *A square, B pub.* The acceptability of the construction is not compromised even in this case. One further argument against compositionality is that if we try to apply it to polymorphemic proper names, we get unacceptable result: Roosevelt has not lived at Roosevelt square, and Columbo has never been to the Columbo pub.

In the second construction, both head and modifier are proper nouns. The only contribution made by the head to the semantics of the phrase is that we know that the thing referred to by the modifier is a member of the group of things referred to by the head, e.g., Volvo S70 is a kind of Volvo, but not a kind of S70.

Regarding polymorphemic proper names in general, we can say that the head *H* bears the semantics of the entire construction, while the only contribution of the modifier *M* is that it shows that *M* is called 'M' and that it is a kind of *H.* This is in contrast with the classic compositional semantics of common nouns, where the *red hat* means a hat which is red, the former president used to be a president, etc., and these implications are non-defeasible.

4.4. Summary

This section gives an overview how we can distinguish between proper names and common nouns using an approach based in linguistics. The first distinguishing property is the unique reference: common nouns, standing by themselves, never have unique reference. They have to be surrounded by other constituents within a phrase to refer some unique entity in the world, while proper nouns have unique reference on their own. There are, however, proper names which seemingly refer to several entities; it is shown through examples that these do have unique reference. Additional linguistic properties of proper names are presented, based on which a stronger distinction between proper names and common nouns can be made. The distinction based on semantic properties is the clearest: common noun phrases are compositional while proper names are not.

5. Conclusion

As can be seen from this overview, the definition of proper names is still an open question in both philosophy and linguistics. If we try to apply the findings presented above to the NER task, we will face various challenges. However, there are a few statements which can be used as pillars of defining what to annotate as NEs.

Early works formulated the NER task as recognizing proper names in general. This generality posed a wide range of problems, so the domain of units to be annotated as NEs had to be restricted. In this restricted domain, we only find person and place names, which have been postulated as proper names from the very beginnings of linguistics (e.g., in Plato's dialogue, *Cratylus*, and in Dionysius Thrax' grammar). The third classical name type, the type of organization names has been mentioned in grammar books from the 19th century. Although the range of linguistic units to annotate was cut, the challenges have remained, since these kinds of names already exhibit properties which make the NER task difficult.

In the expression "named entity", the word "named" aims to restrict the task to only those entities where rigid designators stand for the reference (Nadeau & Sekine 2007). Something is a rigid designator if it designates the same object in every possible world and thus has unique reference – unique in every possible world. Rigid designators include proper names as well as species names, mass terms, natural phenomena and measurement units. These natural kind terms are only partially included in the NER task. The MUC guidelines allow for annotating measures (e.g., *16 tons*) and monetary values (e.g., *100 dollars*), which are rigid designators according to Kripke's theory. Some temporal expressions, typically absolute time expressions, are also rigid designators (e.g., *the year 2017* is the 2017th year of the Gregorian calendar), but there are also many non-rigid ones, typically the relative time expressions (e.g., *June* is a month of an undefined year). Thus, the rigid designator theory must be restricted to keep out species names, mass terms and certain natural phenomena, but must also be loosened to allow tagging relative time expressions as NEs.

If we say that every linguistic unit which has unique reference must be annotated as a NE, we should annotate common noun phrases as well. However, dealing with common nouns is not part of the NER task, so other linguistic properties of proper names and common nouns must be considered to make the distinction between them stronger. The greatest difference is the issue of compositionality. Applying Mill's, Saussure's, and Kripke's theory about the meaninglessness of names, we must conclude that proper names are arbitrary linguistic units, whose only semantic implication is the fact of the naming. Thus, the semantics of proper names is in total contrast with the classic compositional semantics of common nouns, as they are indivisible and non-compositional units. To map it to the NER task: embedded NEs are not allowed, and the longest sequences must be annotated as NEs (e.g., in the place name *Roosevelt square* there is no person name 'Roosevelt' annotated).

There still remain a quite large number of linguistic units which are difficult to categorize. Typically, they are on the border between proper names and common nouns, which is confirmed by the fact that their status varies across languages. We should not forget that the central aim of the NER task is extracting important information from raw text, most of which is contained by NEs. Guidelines should be flexible enough to allow the annotation of such important pieces of information. For getting a usable definition of NEs, the classic Aristotelian view on classification, which states that there must be a *differentia specifica* which allows something to be the member of a group, and excludes others, is not applicable. For our purposes, the prototype theory (Rosch 1973) seems more plausible, where proper names form a continuum ranging from prototypical (person and place names) to non-prototypical categories (product and language names; Langendonck 2007 – consider the parallelism with the order in which names are mentioned in grammar books). Finally, the goal of the NER application will further restrict the range of linguistic units to be taken into account.

References

ACE. 2008. ACE (Automatic Content Extraction) English annotation guidelines for Entities. Version 6.6. Linguistic Data Consortium.

Anderson, J. M. 2007. The grammar of names. Oxford: Oxford University Press.

Chinchor, N. 1998. MUC-7 Named Entity Task Definition version 3.5. In Proceedings of the 7th Message Understanding Conference (MUC-7).

Chinchor, N., E. Brown, L. Ferro and P. Robinson. 1999. Named Entity Recognition Task definition version 1.4.

Cumming, S. 2012. Names. In E. N. Zalta (ed.) The Stanford encyclopedia of philosophy (Winter 2012 edition). Stanford, CA: The Metaphysics Research Lab, Center for the Study of Language and Information, Stanford University.

Deme, L. 1956. Családneveink alaki sérthetetlenségéről [On the morphological sacredness of Hungarian family names]. Magyar Nyelv 52. 365–368.

Ding, R., C. N. Arighi, J.-Y. Lee, C. H. Wu and K. Vijay-Shanker. 2015. pGenN, a gene normalization tool for plant genes and proteins in scientific literature. PLoS ONE 10.

Frege, G. 2000. Ueber Sinn und Bedeutung. In Stainton (2000, 45–64).

Gardiner, A. 1957. The theory of proper names. A controversial essay. Oxford: Oxford University Press.

Grice, H. P. 1975. Logic and conversation. In P. Cole and J. L. Morgan (eds.) Syntax and semantics, vol. 3: Speech acts. New York: Academic Press. 41–58.

Grishman, R. and B. Sundheim. 1996. Message Understanding Conference – 6: A brief history. In Proceedings of the 16th International Conference on Computational Linguistics (COLING). Kopenhagen, 466–471.

Huddleston, R. and G. K. Pullum. 2002. The Cambridge grammar of the English language. Cambridge: Cambridge University Press.

Katz, J. J. 1972. Semantic theory. New York: Harper and Row.

Kenesei, I., R. M. Vago and A. Fenyvesi. 1998. Hungarian. London & New York: Routledge.

Kornai, A. 1994. On Hungarian morphology (Linguistica, Series A: Studia et Dissertationes 14). Budapest: Research Institute for Linguistics, Hungarian Academy of Sciences.

Krallinger, M., F. Leitner, O. Rabal, M. Vazquez, J. Oyarzabal and A. Valencia. 2015. CHEMDNER: The drugs and chemical names extraction challenge. Journal of Cheminformatics 7. S1.

Kripke, S. A. 1981. Naming and necessity. Cambridge, MA & Oxford: Blackwell.

Langendonck, W., von. 2007. Theory and typology of proper names. Berlin & New York: Mouton de Gruyter.

LDC/Linguistic Data Consortium LCTL Team. 2006. Simple Named Entity guidelines for less commonly taught languages. Version 6.5.

Lukács, Á. 2001. Szabályok és kivételek: a kettős modell érvényessége a magyarban [Rules and exceptions: The validity of the double model in Hungarian]. In Cs. Pléh and Á. Lukács (eds.) A magyar morfológia pszicholingvisztikája [The psycholinguistics of Hungarian morphology]. Budapest: Osiris Kiadó. 119–152.

Merchant, R., M. E. Okurowski and N. Chinchor. 1996. The Multilingual Entity Task (MET) overview. In Proceedings of the TIPSTER Text Program: Phase II. Vienna, VA: Association for Computational Linguistics, 445–447.

Mill, J. S. 2002. A system of logic. Honolulu: University Press of the Pacific.

Miller, D., R. Schwartz, R. Weischedel and R. Stone. 1999. Named Entity Extraction from broadcast news. In Proceedings of the DARPA Broadcast News Workshop. Herndon, Virginia.

Nadeau, D. and S. Sekine. 2007. A survey of Named Entity Recognition and Classification. Lingvisticae Investigationes 30. 3–26.

Quirk, R. and S. Greenbaum. 1980. A university grammar of English. Harlow: Longman.

Rosch, E. H. 1973. Natural categories. Cognitive Psychology 4. 328–350.

Saussure, F., de. 1959. Course in general linguistics. New York: Philosophical Library.

Simon, E. 2013. Approaches to Hungarian Named Entity Recognition. Doctoral dissertation. Budapest University of Technology and Economics.

Stainton, R. J. (ed.). 2000. Perspectives in the philosophy of language. A concise anthology. Peterborough: Broadview Press.

Sundheim, B. 1995. MUC-6 Named Entity task definition (v2.1). In Proceedings of the Sixth Message Understanding Conference (MUC6).

Szabó, Z. G. 2008. Compositionality. In E. N. Zalta (ed.) The Stanford encyclopedia of philosophy (Winter 2008 edition). Stanford, CA: The Metaphysics Research Lab, Center for the Study of Language and Information, Stanford University.

Tjong Kim Sang, E. F. 2002. Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In D. Roth and A. van den Bosch (eds.) Proceedings of CoNLL-2002. Edmonton: Association for Computational Linguistics. 155–158.

Tjong Kim Sang, E. F. and F. De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In W. Daelemans and M. Osborne (eds.) Proceedings of CoNLL-2003. Edmonton: Association for Computational Linguistics. 142–147.

Zhu, J., V. Uren and E. Motta. 2005. ESpotter: Adaptive named entity recognition for web browsing. In 3rd Conference on Professional Knowledge Management. 518–529.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 499–527

Dyadic truth

ZOLTÁN GENDLER SZABÓ
Yale University
zoltan.szabo@yale.edu

KEYWORDS

truth
propositions
semantics
topic
relativism

ABSTRACT

Philosophical orthodoxy holds that 'true' is a monadic predicate. I think this view is only halfway correct: there is indeed a monadic truth-predicate in English and other natural languages but this is not the fundamental truth-predicate we use. What can be true *simpliciter* are particular mental states (beliefs, hopes, wishes, etc.) a thinker might be in or particular speech acts (assertions, denials, suppositions, etc.) a speaker might perform. These mental states and speech-acts are truth-apt because they have propositional contents. But propositions are not true *simpliciter* – they are true *of* situations. Thus, the fundamental notion of truth is relational.

1. Simplicity

G. E. Moore held a simple view about the adjective 'good.' He took it to be a monadic predicate expressing a property:[1]

> "For 'good conduct' is a complex notion: all conduct is not good; for some is certainly bad and some may be indifferent. And on the other hand, other things, beside conduct, may be good; and if they are so, then 'good' denotes some property, that is common to them and conduct [...]."

Compelling though Moore's observation may sound, the simple view is too simple: there are good violinists who are pianists without being good pianists, and if goodness were a property shared by all and only what is correctly said to be good, these people would have to both have and lack goodness.[2] This cannot be right, so – appearances notwithstanding – 'good' is not a monadic predicate.

[1] Moore (1903, 2).

[2] The point goes back to Geach (1956).

Moore held the same simple view about 'true': he considered it a monadic predicate expressing a property. This is no surprise, for if we replace 'good' with 'true', 'bad' with 'false', 'indifferent' with 'meaningless' and 'conduct' with 'report' in the quote above we find the point equally compelling. Alas, it also faces the same objection. A double agent sends a report about a person of interest who happens to be an American to both Moscow and Washington, describing him as 'a foreigner'. Here we have a true report sent to Moscow, which is also a false report sent to Washington. Does this show that 'true' is not a monadic predicate expressing the property of truth?

Yes and no. The example indeed shows that applied to reports 'true' is relational: the double agent's report addresses different audiences, and is true relative to the Moscow context but false relative to the Washington context. However, the standard explanation of this fact brings back monadic truth: the report expresses a true proposition in the former context, a different false proposition in the latter. The usual view in philosophy of language today is that there is a monadic truth predicate in English and other natural languages applicable to all and only propositions, and that everything else we correctly call true (sentences, beliefs, theories, reports, etc.) is true *because* it has propositional content, relative to a context, that is true absolutely. Thus, truth is fundamentally monadic.

I think this view is incorrect. There is indeed a monadic truth-predicate in English and other natural languages, but it does not apply to propositions. What can be true *simpliciter* are particular mental states (beliefs, hopes, wishes, etc.) that a thinker might be in or particular speech acts (assertions, denials, suppositions, etc.) that a speaker might perform. These mental states and speech-acts are truth-apt because they have propositional contents. But the propositions themselves are not true or false *simpliciter* – they are true or false *of* situations. The fundamental notion of truth is thus relational – or, at least, so I will argue in this paper.[3]

My view opposes the first plank of the doctrine Herman Cappelen and John Hawthorne have dubbed *Simplicity*. I won't quarrel here with the other four planks: that the semantic values of declarative sentences relative to contexts of utterance are propositions; that propositions are the objects of certain mental attitudes; that propositions are the objects of illocutionary acts; and that propositions are the objects of agreement

[3] I hold a similar view about 'good' – in its core sense it is a predicate expressing a relational property of being good in a way; cf. Szabó (2000a). The Geachean alternative is that 'good' is not a predicate at all, but a predicate-modifier.

and disagreement.[4] I even agree with part of the first plank – I too believe that propositions instantiate fundamental truth and falsity. What I reject is that truth and falsity are properties; I believe they are relations to situations.

Cappelen and Hawthorne spend most of their book warding off challenges against Simplicity coming from those who maintain that a limited number of linguistic devices (epistemic modals, predicates of personal taste, terms of aesthetic or moral approval) are used to build sentences that express propositions whose truth is relative to something (a body of knowledge, a standard of taste, or prevailing opinion). They call such proposals *relativist*, and this is certainly one of the standard ways to use this loaded term. The other standard use is narrower: it requires that propositional truth be sensitive to contexts of assessment.[5] The view I defend differs from the usual relativist views in two important respects. I do not think that the need for relativization is tied to special vocabulary, and I do not propose that truth is relative to something mental or subjective.[6] Rather, I claim that *all* propositions expressed by our declarative sentences are true or false relative to situations. At the same time, I maintain that the proposed view provides a common framework in which these special forms of relativism can be fruitfully debated.

My positive argument has a Quinean flavor despite its distinctly non-Quinean conclusion. It goes as follows. Monadic truth-predicates are ill-suited for the purposes of semantics. If we take semantics seriously, we should either accept that truth is the relation the truth-predicate employed in our overall best semantic theory picks out, or provide some adequate paraphrase of that theory which employs only a monadic truth-predicate. For most standard relative truth-predicates employed in semantics these days such paraphrases can be found. But when it comes to 'sentence S is true at context c and situation s' we can only provide a paraphrase in terms of dyadic propositional truth. I will make a case that an adequate semantic theory does need this truth-predicate, and also that we should accept the paraphrase. If I am right we have good reason to think that propositional truth is dyadic.

[4] Cappelen & Hawthorne (2009, 1). I do, however, reject the claim that mental states or speech-acts are individuated in terms of their objects.

[5] The broader notion is employed by Kölbel (2002), the narrower by MacFarlane (2005).

[6] What opponents of relativism find objectionable tends to be not relativism *per se* – it is the more specific doctrine that truth is relative to what people happen to think or want.

This sort of argument has obvious limitations. Even if our physical theories make reference to numbers and functions it would not follow that there are such things – our theories might be false. Literal falsehood need not be a fatal flaw: the right sort of falsehood can make a theory more perspicuous and more explanatory than any of its available true competitors. Obviously, this could be true for semantics as well: it may well be that semantic theories are committed to propositions being true or false relative to situations and that in virtue of this very commitment they are false, despite providing insightful accounts of the semantic facts. Moreover, unlike physics, semantics is not an established science – it is not beyond the pale to suggest that it is simply on the wrong track. I take both the general fictionalist challenge and the particular concerns about the relatively undeveloped state of semantic theorizing seriously, so I will put my conclusion carefully: to the extent that we think our currently best semantic theories are literally true we have reason to think that propositional truth is dyadic. I hope the thesis is of interest despite the qualification.

2. Context and index

Why can't we use a monadic truth-predicate in semantics? Such a predicate works well as long as we are concerned with the language of the sentential calculus: in saying that $\neg p$ is true just in case p is not, or that $p \wedge q$ is true just in case both p and q are, we are entirely successful in specifying the truth-conditions of certain complex sentences in terms of the truth-conditions of their syntactic constituents. Indeed, I think we can regard semantic theory here as providing an *explanation* why $p \wedge q$ has the truth-value it has. If $p \wedge q$ is true, it is true because both p and q are true, and if $p \wedge q$ is false, it is false because it is not the case that both p and q are true.[7] This is just what semanticists seek to do: they explain why certain complex expressions have the semantic values they do in terms of the semantic values of their constituents and their structure.[8] The prob-

[7] For a defense of this view about the explanatory power of semantic theories, see Szabó (2019).

[8] I would like to stay clear of the debate whether this is *all* semantics should do. There are familiar arguments to the effect that the meaning of a sentence is not exhausted by its truth-conditions, including considerations relating to attitude reports, presupposition, discourse dynamics, and conventional implicature. What matters here is the relatively uncontroversial claim that for declarative sentences difference in truth-conditions is sufficient for difference in meaning.

lem is that to provide this sort of explanation in the case of more complex languages semanticists need relative truth-predicates.

$\mathcal{L}_{\forall}$, the language of the predicate calculus, contains variables – uninterpreted expressions substitutable for interpreted ones. Unlike the individual constant *Viktor*, the variable x is not assigned anything by the interpretation function but it can still replace *Viktor* anywhere *salva beneformatione*. The string *corrupt* (x) is well-formed yet neither true nor false *simpliciter* – it is true relative to an assignment g if and only if g maps x to a member of the set the interpretation function assigns to *corrupt*. This relative truth-predicate is the only one used to articulate truth-conditions in $\mathcal{L}_{\forall}$. For example, *corrupt* (*Viktor*) is true relative to an assignment g if and only if whatever the interpretation function assigns to *Viktor* is a member of whatever it assigns to *corrupt*.

Terminology can mitigate discomfort: we can introduce the term 'formula' to refer to a category of expressions that include both *corrupt* (*Viktor*) and *corrupt* (x) and reserve the term 'sentence' for formulae without free variables. Then we do not have to say that sentential truth in $\mathcal{L}_{\forall}$ is relative to assignment. Absolute truth can be introduced through a meaning postulate: if φ is a sentence then φ is true iff φ is true relative to all assignments. But verbal magic does not change the facts: insofar as our concern is to account for the truth-conditions of sentences of $\mathcal{L}_{\forall}$, this new monadic truth-predicate is completely idle. The explanation of why a sentence has the truth-value it does proceeds as follows: first we give a complete explanation of why the sentence has the truth-value it does relative to all assignments, and then we tuck the definition of monadic truth to the end.

$\mathcal{L}_{\Box}$, the language of the modal sentential calculus, has intensional operators – expressions whose syntactic profile is to yield sentences when combined with sentences, and whose semantic profile is undefinable via truth-tables. Ascription of truth-conditions to $\Box$ (*corrupt* (*Viktor*)) relies on the truth or falsity of *corrupt* (*Viktor*) relative to possible worlds: the sentence is true relative to a possible world w just in case it is true relative to all possible worlds accessible from w. This relative truth-predicate is employed in ascribing truth-conditions to *corrupt* (*Viktor*) as well: the sentence is true relative to a possible world w if and only if what the interpretation function assigns to *Viktor* at w is a member of what it assigns to *corrupt* at w.

Once again, there is a terminological move that can make the departure from our usual talk of absolute truth seem less drastic. We can distinguish a world among all the possible ones, and call it actual. With this extra machinery in hand we can define absolute truth as truth relative to the actual world. But again, as far as the semantics of $\mathcal{L}_{\Box}$ is concerned, the absolute truth-predicate is a fifth wheel. The explanation of why a sentence has the truth-value it does proceeds as follows: first we give a complete explanation of why the sentence has the truth-value it does relative to an arbitrary possible world, and then we apply this general explanation to the actual world.

If we have both individual variables and intensional operators – as in the language of quantified modal logic, $\mathcal{L}_{\forall\Box}$ – we need a truth-predicate that is relativized both to assignment functions and possible worlds. The reason the two relativizations cannot be collapsed into one is simple: variables do not shift their semantic values when they occur within the scope of intensional operators. Whether $\Box\,(corrupt\,(x))$ is true at an assignment and a world depends on the truth or falsity of $corrupt\,(x)$ at the *same* assignment at all the *different* worlds; whether $\forall x\,(corrupt\,(x))$ is true at an assignment and a world depends on the truth or falsity of $corrupt\,(x)$ at the *same* world at all the *different* assignments.[9] We have independent sources of variation in the truth-value of $corrupt\ (x)$.[10]

Natural languages contain expressions that may be considered as variables or intensional operators. Third person singular pronouns are fairly uncontroversial examples of the former, modal auxiliaries of the latter.[11] 'He is corrupt' is not true *simpliciter* – it is true only relative to assignments that map the pronoun 'he' to a male person who is corrupt. The truth-conditions of 'Viktor must be corrupt' are not fixed by the truth or falsity of 'Viktor is corrupt' – they are determined by the truth or falsity of 'Viktor is corrupt' relative to possible worlds accessible from actuality.

There are also expressions in natural languages that are pretty clearly not variables or intensional operators, but share enough with them to warrant analogous semantic treatment. First and second person singular pronouns are not variables, for normally they cannot be bound by a quan-

9 Cf. Lewis (1980).

10 Truth-predicates are also relativized to models. Models are necessary to define logical consequence, but not for the semantics – they provide *alternative* interpretations for the same expressions. Thus, I will ignore relativization to models it in what follows.

11 This is not to say that treating pronouns as variables or modal auxiliaries as intensional operators is obligatory. There are successful semantic theories on the market that interpret pronouns as identity-functions (cf. Szabolcsi 1987 and Jacobson 1999) and modal auxiliaries as quantifiers (cf. Percus 2000 and Keshet 2008).

tifier.[12] Yet they are variable-like insofar as linguistic conventions alone fail to determine their referent. It is customary to think that all variable-like expressions receive their semantic value somehow or other from the *context of utterance*.[13] The expression 'necessary' is not an operator, for it cannot directly combine with a sentence. But if we choose to interpret 'must' as an operator, we should probably seek an operator-like interpretation for 'necessary' as well, so as to account for their tight semantic connection. It is customary to lump all the information needed to interpret operator-like expressions into an *index of evaluation.* Semantic theories of sufficiently large fragments of natural languages use a truth-predicate relativized to both context and index. They do not employ unrelativized truth-predicates at all.

If truth is fundamentally monadic, the relational truth-predicates of semantics must somehow be analyzed in terms of a monadic one. If that cannot be done we would have good reason to believe that truth is the very relation picked out by the irreducible truth-predicate employed in explaining the truth-conditions of sentences of our languages in our best semantic theories.

3. Defining the relativized truth-predicate

How can we analyze 'sentence S is true at context c and index i' using a monadic truth-predicate? First, we need something monadic truth-predicates might plausibly apply to. Sentences won't do – 'He is corrupt' is not true or false *simpliciter* only relative to some value context provides for the pronoun. However, we might conjecture that by assigning an individual to the pronoun (e.g., Viktor) context associates a proposition with the entire sentence (e.g., the proposition that Viktor is corrupt) and that the truth of the sentence relative to the context can be defined as the truth of the

[12] There are bound readings much discussed in the linguistic literature, such in 'You are the only one who eats what you cook'. For an extended discussion of the state of the art on these "fake indexicals", see Kratzer (2009).

[13] It is not customary to say that the assignment function is a feature of the context of utterance. Kaplan (1989) treats indexicals (including demonstratives) as constants and regards the assignment function as a parameter distinct from both context and index. This has the disadvantage of flouting the ideal of uniform interpretation for pronouns. Contemporary semantic approaches typically view all pronouns as variables and regard the assignment function as a parameter of the index; cf. Heim & Kratzer (1998). It is the context of utterance that initiates the assignment function of the index, which can then be shifted when quantifier expressions are evaluated.

associated proposition. Propositional truth is still index-sensitive but that can be captured using a subjunctive conditional and an appropriate indexical. 'Proposition p is true relative to possible world w' can be defined as 'if possible world w were actual proposition p would be true.'[14] The doubly relativized truth-predicate of semantics can then be defined as follows:

(1) Sentence S is true at context c and index w if and only if S expresses a proposition in c that would be true if w were actual.

The main line of opposition to (1) in contemporary philosophy of language comes from those who maintain that propositions are never, or hardly ever, expressed in language.[15] In their view, what most declarative sentences express are *incomplete* entities, things that become propositions only when properly supplemented. A variety of terms have been floated for these entities: 'propositional fragments', 'propositional skeletons', 'propositional radicals', 'propositional frames', etc. So, for example, one might hold that the sentence 'Andy is rich' expresses a propositional function f from amounts of money to propositions, such that for any amount a, $f(a)$ is the proposition that Andy has wealth in excess of a. There are many other examples cited in the literature – 'Árpád is subservient' might express a propositional function that assigns to a class of individuals c the proposition that Árpád is subservient to members of c, 'Lőrinc's contract is illegal' might express a propositional function that assigns to a relation r the proposition that the contract that bears r to Lőrinc is illegal, and so on.

I think this view rests on an overly restrictive conception of context. I will illustrate my point using the sentence 'László's mustache is huge' but the considerations apply generally.[16] Imagine someone uttering 'László's mustache is huge' in the course of a conversation about László's latest clash with nosy reporters shown on television. Let's suppose by uttering this sentence this speaker manages to assert the proposition that László's mustache is significantly larger than size s. If so, she asserted this without any obvious indirectness, and accordingly, it seems theoretically parsimonious to say that on this occasion in this speaker's mouth this sentence

[14] The wording in Soames (2010) is different: he defines 'proposition p is true relative to possible world w' as 'p would be true if w were instantiated'. But since Soames thinks possible worlds are properties and since he would cash out 'w is actual' as 'w is instantiated,' this is an equivalent definition.

[15] For classic examples of this view, see Bach (2001) and Carston (2002).

[16] The considerations in the next three paragraph are spelled out in more detail in Szabó (2010).

expressed the very proposition the speaker asserted. Why say instead that the speaker expressed something less, to wit, the propositional function that maps arbitrary sizes to the proposition that László's mustache is larger than that size? The idea must be that the relevant size is not fixed by the context, only by the relevant intentions of the speaker. But why think that those intentions cannot be part of the context? If the speaker managed to assert that László's mustache is significantly larger than size *s* then she must have made her intentions to talk about *s* manifest somehow, and if she did, the fact that she has a particular size in mind became part of the common ground relative to which the sentence must be interpreted.[17] As long as we think of context as common ground it is reasonable to maintain that context determines the comparison class left unarticulated in the sentence 'László's mustache is huge'.[18]

One might doubt that the description of the scenario I gave is plausible. Maybe by uttering 'László's mustache is huge' a speaker can never really assert anything as specific as the proposition that László's mustache is significantly larger than size *s*. This is a fair concern, but it does not save the objection against the idea that 'László's mustache is huge' is true in a context just in case it expresses a true proposition in that context. If we think it is hard, or even impossible to assert a determinate proposition uttering 'László's mustache is huge' because the sentence lacks sufficient specificity then, we should also think it is hard or impossible for the sentence to be determinately true.[19]

[17] See Stalnaker (1998) for a discussion of the fact that indexical expressions must be interpreted not against the context as it was prior to the utterance but as it is after the context is already updated with the information that the utterance has already been made.

[18] I assume that the fact that many philosophers represent context, à la Kaplan, with an ordered *n*-tuple has also contributed to the idea that context cannot provide all the information necessary for identifying the proposition people normally express by uttering various context-sensitive sentences. As Lewis (1980) has observed a long ago, context-sensitivity in natural languages goes way beyond 'I', 'here' and 'now' and so its sources cannot be captured by a simple list of parameters.

[19] One might concede that if we use a sufficiently generous conception of context, declarative sentences do express propositions relative to context, but insist that they do not express them *semantically*. Thus, the relational truth-predicate 'sentence S is true at context c and index i' would belong not to semantics, but to a broader enterprise – truth-conditional pragmatics; cf. Recanati (2010). While I will continue to call this theory 'semantics' those who prefer to think of it under a different label are welcome to rephrase my argument accordingly.

I think the standard view in semantics that assumes that sentences express proportions relative to contexts is perfectly reasonable. The problem with (1) is not that it is false, but rather that it is not sufficiently general. It defines 'sentence S is true at context c and index i' only if we assume that indices of evaluation comprise nothing beyond a possible world. What if we have intensional operators in natural languages that are not modal? 'Soon' and 'nearby' are possible candidates – if they are operators, this is what their semantic clauses might look like:

(2) If σ is a sentence, then *Soon* σ is true at context c and index $\langle w, t, l\rangle$ if and only if there is a time t' in the near future of t such that σ is true at c and $\langle w, t', l\rangle$.

(3) If σ is a sentence, then *Nearby* σ is true at context c and index $\langle w, t, l\rangle$ if and only if there is a location l' in the vicinity of l such that σ is true at c and $\langle w, t, l'\rangle$.

To define truth relative to a context and an index comprising a world, a time, and a location, defenders of Simplicity can extend the blueprint provided by (1). At first, this seems easy:

(1′) Sentence S is true at context c and index $\langle w, t, l\rangle$ if and only if S expresses a proposition in c that would be true if w were actual, t were present, and l were local.

But there is a problem with this suggestion. We have a clear grip on what would be the case if October 25, 1963 were present. The US and the Soviet Union would be entangled in the Cuban Missile Crisis and people in the know would be wondering whether they live another day. By contrast, it is not clear what would be the case if Melbourne were local. Would it be fall, as it is in Melbourne or would it be spring, as it is locally? (Yes, I am writing this in May in the Northern hemisphere.) There seems to be an indeterminacy here, yet it is determinate that 'It is fall' is true at $\langle w, t, \text{Melbourne}\rangle$, as long as w is the actual world and t the present time. Thus, (1′) is by no means a satisfying paraphrase for 'sentence S is true at context c and index $\langle w, t, l\rangle$'.

The problem gets worse if we leave English behind and consider its extensions. Let's say that 'somemoney' as a one-place sentential operator whose semantic clause goes as follows:[20]

(4) If σ is a sentence, then *Somemoney* σ is true at context c and index $\langle w, t, l, ¤\rangle$ if and only if there is a currency $¤'$ legally convertible from $¤$ such that σ is true at c and $\langle w, t, l, ¤'\rangle$.

[20] As I learned writing this paper, '¤' is the currency symbol used when the specific symbol of a particular currency is unavailable.

It seems plausible that such a stipulation can bestow meaning upon 'somemoney'. If you know that one dollar is legally convertible to .89 euros and you know that the latte you just bought in a Starbucks in Manhattan cost you $3.65 you can also tell (perhaps using a calculator) that 'Somemoney, a latte in New York costs €3.25' is true. You also know (without knowing anything about exchange rates) that 'Somemoney, a latte in New York costs $3.65' and 'Somemoney, snow is white' are also true and that 'Somemoney, a latte in New York costs £0' and 'Somemoney, snow is black' are false. All this knowledge suggests that you have acquired competence with this new word. Yet, it seems clear that defining propositional truth relative to currency cannot follow the blueprint. The obvious suggestion – proposition p is true at currency ¤ just in case p would be true if ¤ were a local currency – fails spectacularly. Plausibly, if the Euro were a local currency in New York then the US would be a member of the EU, yet there is nothing in clause (4) that would suggest that 'Somemoney, the US is a member of the EU' is true just because the latte you just bought in a Starbucks in Manhattan cost you $3.65. There seems to be no hope to paraphrase the relative truth-predicate 'sentence S is true at context c and index $\langle w, t, l, \text{¤}\rangle$' in terms of absolute truth.

To fend off these objections, defenders of monadic truth have to deny the need for indices beyond worlds and times. They would have to argue that despite the explicit stipulation, we understand 'somemoney' as a quantifier over currencies. Thus, instead of trying to use (4) to interpret 'Somemoney, a latte in New York costs €3.25', we paraphrase this sentence as 'There is some currency ¤ such that the price of a latte in New York in ¤ is legally convertible to €3.25' which we can understand perfectly well. Our ability to provide such a paraphrase comes from understanding (4) by analogy and from our general capacity to articulate sentences in English which manifest this understanding. No need then to assume that indices could include currencies. Of course, these are hefty claims about the way we in fact purse strings containing 'somemoney' – that is, bold hypotheses about human psychology.

Making psychological assumptions is enough when it comes to made-up words, like 'somemoney' but to defend the simplicity of indices in light of our proposed semantics for 'nearby' defenders of monadic truth will have to descend into the trenches of linguistic semantics. They would argue that (3) is not a correct semantic clause because 'nearby' is a quantifier that binds location variables. This claim raises a host of questions about location variables. Are they base-generated or traces left behind after movement? What happens to them when there is no expression like 'nearby' to bind them? What is their semantic type? Why is there no expression that is used to articulate them phonologically in English? Is there such an expression in other natural languages?[21] These are substantive empirical questions about which there is considerable disagreement among the experts.[22]

Here is where we stand. I argued that semantics needs the relational truth-predicate 'sentence *S* is true at context *c* and index *i*'. If truth is fundamentally monadic, we should be able to analyze this predicate in terms of monadic truth. I claimed that relativity to contexts is indeed analyzable in this way: if we set aside controversial cases, the claim that a declarative sentence is true at a context just in case it expresses a true proposition at that context is quite plausible. Relativity to indices is more problematic. If indices contain nothing more than a possible world and perhaps a time, truth at an index can be analyzed by means of an appropriate counterfactual, but such an analysis is not available for richer indices. Thus, defenders of the idea of that truth is fundamentally monadic are forced to say that natural languages contain no operators, except perhaps modal and temporal ones. It is a mark against the traditional view that it is forced to take a strong stand on unresolved empirical questions but I concede that this is not a decisive argument against it. In the next section, I consider a different line of attack.

4. Topic sensitivity

There is an old idea, going back at least to J. L. Austin's 1950 paper on truth, according to which the statement one makes in uttering a sentence is true just in case the situation the statement is about is of the type identified by the meaning of the sentence.[23] The benefits of thinking along

[21] For a thorough discussion of the question whether we should postulate location variables in natural languages, see chapter 3 of Recanati (2010).

[22] It is sometimes suggested that we should try to avoid postulating variables in syntax, whenever possible. This would not fully resolve the operator vs. quantifier debates but it would give the upper hand to proponents of operators. But those who are willing to follow this methodological principle should eschew variables altogether – after all, we do have variable-free semantic theories that fare rather well in accounting for the truth-conditions of English sentences.

[23] "A statement is said to be true when the historic state of affairs to which it is correlated by the demonstrative conventions (the one to which it 'refers') is of a type with which the sentence used in making it is correlated by the descriptive conventions" Austin (1950/1961, 122).

these lines are illustrated by an example due to Jon Barwise and John Etchemendy (with names changed slightly for ease of cognitive load):[24]

> "We might imagine, for example, that there are two card games going on, one across town from the other: Xavier is playing cards with Anna and Beth, and Claire is playing cards with Dana. Suppose someone watching the former game mistakes Anna for Claire, and claims that Claire has the three of clubs. She would be wrong on the Austinian account, even if Claire had the three of clubs across town."

Let us call the bystander Yolanda and let us assume that she makes her statement by uttering the sentence (5):

(5) Claire has the three of clubs.

Yolanda's statement appears to be untrue. (I leave the question open whether it is false or simply lacks a truth-value.) Now imagine that across town Zoe is watching Claire's game and at the same time also utters (5). That statement is undoubtedly true. This pair of observations is the data to account for. The Austin-inspired line is as follows: in uttering the same sentence, Yolanda and Zoe stated the same thing (i.e., expressed and assented to the very same proposition) but made different statements (i.e., performed different assertions). What they both stated was the proposition that Claire has the three of clubs. They made different statements because they were concerned with different situations (call these the *topic* situations) when they stated that Claire has the three of clubs. If this is the right way to think about the case, the proposition that Claire has the three of clubs is *topic-sensitive* – its truth-value depends on which situation is the topic the speaker asserting the proposition is talking about.

The Austinian view gives up the simple assumption that we can individuate speech-acts and mental states by their contents. Yolanda and Zoe assert and believe the same thing – the proposition that Claire has the three of clubs. Yet Zoe's assertion and belief is true, while Yolanda's is not. To say what their assertions and beliefs *are* besides their contents, one must rely on their topics as well. Yolanda asserts and believes that Claire has the three of clubs regarding the game Anna and Beth are playing, Zoe asserts and believes that Claire has the three of clubs regarding the game Claire and Dana are playing.

There are more conventional alternatives to the Austinian line and I am fully aware of the fact that they are *prima facie* more attractive to

[24] Barwise & Etchemendy (1987, 122–123).

many. I will try to argue that these alternatives face difficulties that make them in the end less appealing. But before I try to do that I'd like to make the case that *if* the Austinian account of the example is correct then we are stuck with an irreducibly dyadic propositional truth-predicate.

Recall that we can define 'proposition p is true at possible world w' as 'p would be true if w were actual'. To define 'proposition p is true at situation s' analogously we would need to replace 'actual' by an appropriate indexical for situations. In fact, we have no such indexical in English but we could perhaps introduce one by *fiat.* Let's stipulate that 'topical*' refers in any context to the topic situation of the context. (The star is there to distinguish this freshly minted word from the English 'topical'.) Then we could try the following definition:

(6) The proposition p is true at the situation s iff p would be true if s were topical*.

Does this work? It might if 'topical*' behaves just like 'actual' does within the antecedents of subjunctive conditionals. But not all indexicals do – 'local' seems like a counterexample, given the fact that 'It would be winter here if Melbourne were local' does not seem to be determinately true or false. This suggests that antecedents of subjunctive conditionals cannot shift the location against which the consequent is evaluated in the way in which they can shift the world.

Is there a definition in English? I can think of one plausible candidate which exploits the intuition that situations are *parts* of the world. The idea is that truth at a situation is nothing more than truth at a situation-sized world:

(7) The proposition p is true at the situation s iff p would be true if s were the actual world.

According to (7) the proposition that Claire has the three of clubs is true at the card game between Claire and Dana because if that card game were all there is to actuality Claire would indeed have the three of clubs. But the proposition is not true at the card game between Anna and Beth because if the actual world were just that card game, Claire would not have the three of clubs (indeed she would not even exist).

While this might be acceptable in the case at hand, it fails in general. Consider the sentence 'I do not exist' and imagine that Yolanda utters it while she is talking about a card game in which she is not a participant. (7) predicts that she is clearly speaking the truth: after all, if that card game had been all there is to actuality Yolanda would indeed fail to exist. But that is counterintuitive – maybe she speaks falsely, maybe her statement is neither true nor false, but it is surely not straightforwardly true. Could we say that topic situations must always be big enough to contain everything the speaker is referring to? That might explain why 'I do not exist' cannot be used to make a true statement. But the explanation is not particularly plausible – if Yolanda uttered 'Only four people exist' or 'An hour ago there was nothing' or 'Any two things are at most a few yards apart' she would not be speaking truly despite the fact that she would not be referring to anything outside the card game.

Is there some way other than (6) or (7) to define truth at a situation in terms of monadic truth? I cannot prove that there is not but I certainly do not know of any. If we need to use 'sentence S is true at context c and situation s' in the semantics of natural languages then I think we should concede that we are employing a truth-predicate that we cannot define in terms of a monadic propositional truth-predicate.

5. Against invariantism and contextualism about topic-sensitivity

The question remains whether we really need the truth-predicate 'sentence S is true at context c and situation s' in the semantics of English and other natural languages. Is the Austinian account of the statements made by Yolanda and Zoe correct?

The example of Claire and the three of clubs has been around for a while and it has failed to convince most semanticists that propositional truth is relative to topic situations. There are two main lines of resistance: the *invariantist* and the *contextualist* one. The invariantist denies the existence of topic-sensitivity, claiming that (5) has the same truth-value as uttered by Yolanda and Zoe. The contextualist accepts topic-sensitivity and accounts for it by claiming that Yolanda and Zoe express different propositions. I think there are strong reasons to reject both of these views.

The invariantist will point to the fact that while Yolanda's claim is infelicitous it does not seem outright false. Perhaps we find it infelicitous because we are told about her mistaking Anna for Claire and hypothesize that she did not really mean what she said. (When she recognizes her mistake, she may indeed say 'Oh, I did not mean that – Ann has the three of clubs.') If this is the reason we find Yolanda's utterance odd, its infelicity is independent of its truth-value. So, maybe (5) is actually true when uttered by Yolanda.

This response can be disarmed by changing the example. Suppose that Claire is simultaneously playing two on-line card games and she has the three of clubs in one but not the other. Yolanda follows the second game on a screen and knows nothing about the first. She does not know what cards Claire holds but makes a bet uttering (5). It seems perfectly clear that Yolanda loses this bet. Her bet concerns the game in which Claire does *not* have the three of clubs, it is not based on any misidentification, and there is no plausibility to the claim that she somehow failed to say what she meant.

Still, invariantists could insist that we should distinguish between the proposition expressed by Yolanda's utterance and the proposition asserted by her. The idea would be that the job of semantics is nothing more than to associate, based on linguistic conventions in a context-independent way, a proposition with sentences. So, (5) expresses the proposition that Claire has the three of clubs, even though what Yolanda meant *and* said was a different proposition, to wit, that Claire has the three of clubs *in the game she is following*. This latter proposition is obtained from the proposition expressed through a pragmatic process called *enrichment*.[25]

The trouble is that there is an element of ineliminable *arbitrariness* in the semantic project thus construed, as long as we take propositional truth to be monadic. Is the proposition that Claire has the three of clubs true or false when Claire has the three of clubs in one on-line card game but not the other? Many would say it's true, on the account that she does have the three of clubs in *some* ongoing game. But why not say instead that it's false because she does not have the three of clubs in *every* ongoing game? If our semantic project is supposed to abstract away from the vagaries of context there seems to be no good reason to prefer the first option to the second. The pure linguistic meaning of 'Claire has the three of clubs' seems neutral on how many games she is supposed to have the three of clubs in – the sentence does not encode existential, universal, or any other kind of quantification over card games. The sensible way to avoid the arbitrary choice is to concede that the proposition that Claire has the three of clubs is true at one game but false at the other.[26] But once we come this far,

[25] The debate on modulation is voluminous. For a classic attack on the idea, see Stanley (2000; 2002); for a classic defense, see Recanati (2002; 2004).

[26] One might avoid arbitrariness by pleading ignorance: the proposition that Claire has the three of clubs is determinately true or false when she has that card in one game but not in another, we just don't know which. This is the sort of view advocated by Cappelen & Lepore (2005). But if we really understand this proposition why can't we tell whether it is true in the simple case described? Is there some information we are missing? The proposal has much in common with the view that vague sentences express propositions whose truth-value we cannot know. Except that in the vagueness case there is a story about the source of the ignorance and here there is none.

the motivation for denying that the proposition expressed by the sentence Yolanda uttered is exactly what she said and meant evaporates.

Contextualists accept that (5) is topic-sensitive but argue that this is so simply because it expresses different propositions on different occasions. In their view, in uttering (5) Yolanda and Zoe both asserted the proposition the sentence expresses in their respective contexts. In the context Yolanda was in this was the proposition that Claire has the three of clubs in *s*, where *s* is the situation Yolanda was talking about; in the context Zoe was in, it was the proposition that Claire has the three of clubs in *s*', where *s*' is the situation Zoe was talking about. Thus, the proposition Zoe asserted is true *simpliciter* while the one Yolanda asserted is not. One reason this line may appear promising is that (5) contains the definite description 'the three of clubs'. On the Russellian view, this is a quantifier phrase which, in a plausible semantic theory, is associated with a domain. If you think what situation a speaker talking about fixes the domain of the description, you immediately predict that Yolanda and Zoe expressed different propositions.[27]

The semantics of definite descriptions and the pragmatics of domain choice are complicated and philosophers of language have strong feelings about them. I do too and I'd rather not go into this here.[28] Fortunately, we can change the example and bypass the issue. Suppose that instead of (5) Yolanda and Zoe had uttered (8), and suppose that in the game Zoe is observing Claire indeed has a strong hand:

(8) Claire has a strong hand.

Since the predicate is an idiom, there is no overt element in this sentence that could be construed as a quantifier in need of a domain. Of course, there might be covert elements; contextualists have every right to hypothesize that (8) expresses the proposition that Claire has strong hand in *s*, where *s* is a contextually supplied situation. But there is no *independent* motivation for this, beyond the desire to keep propositional truth

[27] One should not think that topic situations *always* fix quantificational domains. Normally when one utters a sentence like 'The researchers monitored everyone's sleep' one is talking about a situation that includes some sleeping experimental subjects and some wide-awake researchers. Yet, the statement can be true. The obvious suggestion is to let context assign to 'everyone' a restricted domain, thus guaranteeing that the sentence in context expresses the proposition that the researchers monitored the sleep of every experimental subject. This proposition will then be true at a situation *s* where the researchers in *s* monitor the sleep of every experimental subject in *s*.

[28] See Stanley & Szabó (2000) and Szabó (2000b; 2005).

monadic. I will discuss contextualism using (5) but if the presence of the definite description distracts you, feel free to replace it with (8).

I was a contextualist about topic sensitivity for a long time. But then I noticed that the view has a really bad consequence. In the original example, the topic situation is the one Yolanda is observing – a situation involving Anna and the particular cards she holds in her hands. Since Anna could not be Claire, it appears that this particular situation could not be one in which Claire has the three of clubs. In the modified example, the topic situation is the one represented on the computer screen Yolanda is observing – a situation involving Claire and the particular cards she has in a game. Since none of those cards is the three of clubs, it seems that this particular situation also could not be one in which Claire has the three of clubs. So, in both cases, what Yolanda said cannot be true.[29] But this is intuitively wrong: in both examples, what Yolanda said was false but could have been true. The problem with contextualism is that it construes the proposition that Claire has the three of clubs as the position that *s* – the particular situation the speaker is talking about – is such that Claire has the three of clubs in it, and this proposition is not a contingent one.

This argument relies on a metaphysical assumption, to wit, that a situation involving someone and some cards could not be identical to a situation involving someone else or some other cards. This can be challenged. One might say that the situation Yolanda is talking about in the original example is a particular card game where Anna plays but *that very card game* could have been one where Claire plays instead, and that the situation Yolanda is talking about in the modified example is a particular card game where Claire does not have the three of clubs but *that very card game* could have been one where she does. Maybe so. But no matter who the players are and what cards they hold, these situations would still be *card games.* Thus, if Yolanda utters 'A card game is going on' concerning either of the situations, she speaks the truth. And if the proposition she asserts in uttering this sentence is that a card game is going on in *s*, where *s* is the topic situation in the context of utterance, then what she asserted would be necessarily true. And this is still deeply counterintuitive.

Contextualists could avoid the troubling modal commitments by going *descriptivist* about the topic situation. Thus, they could say that the

[29] It is no good to insist that in some epistemic sense Ann could be Claire, and a card that isn't in fact the three of clubs could be the three of clubs. This is true, and consequently it is also true that what Yolanda asserted could be true is some epistemic sense. It is still predicted to be metaphysically necessary, which is bad enough.

proposition Yolanda expresses is not the proposition that Claire has the three of clubs in *s*, where *s* is just a variable whose value is the particular game she is observing but rather the proposition that Claire has the three of clubs in *d*, where *d* is a definite description picking out the situation she is observing. Then, assuming the description is well chosen, there will be possible worlds where *d* picks out a different situation or no situation at all, and the contingency of the proposition expressed is secure.

But I do not think this approach can capture the relevant intuitions. Consider some suggestions about what the missing description might be. It could be something like 'the game I am observing' or 'the game going on at that table'. Is the proposition Yolanda expressed contingent because she could have been observing a different game, or because the game she is talking about could have been going on at a different table? Hardly. The intuition is that her statement is *de re* – its topic is a particular situation, the one she is attending to, not some situation or other that fits the way in which she might describe this situation. The puzzle is how it can be still contingently false, given that it characterizes that situation as being a way it could not be. The Austinian view solves the problem by separating topic from content: Yolanda's statement is *de re* but the content of this statement is contingent (true at some situations but not of others).

I accept the Austinian account of our key example. And while this is just a single example, I also believe it is fairly clear that topic-relativity is a general phenomenon. Usually when someone makes an assertion we can ask them to identify the situation they are talking about.[30] When such a request sounds most unreasonable (e.g., when someone utters 'Snow is white' or 'Unicorns do not exist' or 'There is no largest prime') it can still be answered by saying that the topic of one's assertion is the whole world. In claiming that propositional truth is dyadic, we avoid the problems invariantists and contextualists are stuck with. Unlike invariantists we are not forced to make an arbitrary choice about the truth-value of certain propositions, and unlike contextualists we do not have to deny their contingency.

Accepting topic-sensitivity for propositions is not a semantic theory; it's just a constraint of how semantic theories should be constructed. The particular way topic-sensitivity is usually built into semantic theories by those who believe in it is contrary to my own view. Situation semantics, motivated in part by the very example I cited, distinguishes two levels

[30] Of course, uttering 'What situation are you talking about?' may not be the best way to ask this question. We can think of follow-up questions involving 'where' or 'when' or 'which' as aiming at specifying the topic situation.

of content for a sentence in context: the *infon* (roughly, what I called the proposition the sentence expresses in the context) and the *Austinian proposition* (something that comprises both the infon and the topic situation of the context).[31] The Austinian proposition is supposed to be true *simpliciter* just in case its infon is true relative to its topic situation. However, it does not seem like semantics is in need of two different entities playing the role of content for each declarative sentence. Once you have the infon, you have everything you need for explaining truth-conditions. Austinian propositions have nothing to do except to ensure that there is some content that can be true or false *simpliciter.*

But do not we have content that is true or false *simpliciter* anyway? Let p be the proposition that Claire has the three of clubs and let s be the situation across town including Claire holding the three of clubs. Then p is true at s. There is also the proposition p' that p is true at s. Isn't p' a proposition that is true *simpliciter*? I don't think so. Unlike p, which is true at some situations and false at others, p' is true at all situations (or at least, it is not false at any). The difference between p and p' is that the former is contingent and the latter is not, but this does not affect the fact that their truth is equally relative to situations. There is, of course, Zoe's assertion and the belief she expresses when she utters 'Claire has the three of clubs' talking about s, and these are indeed both true *simpliciter.* But these are representations, not contents of any sort. Representations can be true *simpliciter*, provided their contents are true at the situation they are about.

A common objection against the sort of view I recommend is that it fails to respect the intuition propositions must be *complete.* The charge is that whenever we find ourselves lured into thinking that a proposition is true relative to something or other, that's a clear sign that we are not really thinking of a proposition, only a propositional function.[32] But what is the relevant notion of completeness? We do have intuitions about certain sentences being syntactically incomplete: 'Claire does too' is a well-formed sentence, but without knowing the antecedent of 'too' we perhaps cannot know *which* sentence it is. We also have intuitions about certain sentences being semantically incomplete: 'Claire is ready' is a meaningful sentence, but without knowing what Claire is ready for we perhaps cannot know *what*

[31] See Barwise & Etchemendy (1987) and Barwise (1989).

[32] Recanati (2008) advocates a version of standard situation semantics against the view I defend (which he labels "radical relativism") on the basis of this objection.

it means. These intuitions can be criticized but I think we are better off respecting them.[33] However, 'Claire has the three of clubs' and 'Claire has a strong hand' appear to be complete in both of these senses. The complaint that this sentence fails to express a complete proposition bottoms out in the observation that without knowing which of the simultaneous games we are talking about we cannot know whether it is true *simpliciter*.[34] Since I do not think they are true *simpliciter*, this does not move me.

Let me summarize the main argument of the paper. In this section, I have argued that declarative sentences are topic-sensitive and semantic theory should employ a relativized truth-predicate 'sentence S is true at context c and index i', where i includes a topic situation. In the two sections before this one, I argued that while this predicate can be analyzed as 'sentence S expresses at context c a proposition that is true at index i', if index i includes a situation this cannot be further analyzed in terms of monadic propositional truth. The conclusion is that unless our semantics is on the wrong path, we have reason to think that truth is not a property of propositions but a relation they bear to situations and to whatever else is included in the indices. In the next section, I will argue that indices needn't contain anything other than situations. This rounds up the case for my central claim: that truth is a relation between positions and situations.

6. Worlds, times, events, and propositions

I favor a conservative way of building topic sensitivity into the semantics: simply replace possible worlds with possible situations in giving truth-conditions for logically simple declarative sentences. This is a minimal change as far as the basic structure of the theory is concerned. There are dif-

[33] Chapter 5 of Cappelen & Lepore (2005) contain skeptical arguments against appeals to incompleteness.

[34] The classic place for voicing such concerns is Evans (1985). Evans argues against temporally neutral propositions and points out that assertions made in uttering a tensed sentence "would not admit of stable evaluation as correct or incorrect" (349). To my mind, this conflates two senses of 'assertion'. What one asserts in uttering 'Socrates is sitting' can be true at one situation yesterday and false at another today, so they can indeed not be evaluated *tout court*. But the act of assertion (or the particular belief expressed when that act is performed) which is about a particular situation is true or false *simpliciter* depending on whether that situation is one where Socrates is sitting or one where this isn't the case.

ferent ways to expand such a semantics to logically complex declarative sentences.[35]

What are situations? I will not be able to give a particularly informative answer, for I think situation is a basic ontological category. What I can do is argue that our understanding of 'situation' is no worse than our understanding of 'object'.

The technical term 'object' designates all the paradigm objects (tables, chairs, coffee cups), things people would occasionally call objects (mountains, animals, people), and then also some things people would never call objects (clusters of galaxies, centuries of time, theorems of mathematics). It is doubtful that these things share some language-independent characteristic. We can say, following Frege, that objects are the things we designate with singular definite descriptions. And since singular definite descriptions are built from count nouns (possibly modified by adjectives and relative clauses) we can say that objects are the things to which count nouns apply.

Situations can be topics, that is, they are the sorts of things that comprise what we are talking about. Claire's playing a game of poker is a paradigm situation, Claire's betting $5 in a poker game is something people would occasionally call a situation, and Claire's cheating in her Thursday evening poker games over the course of a decade would probably never be called a situation. Yet, I want to use the word 'situation' in such a broad way as to cover all these and much else. The possessive constructions I listed are usually treated in semantics as singular definite descriptions built around a gerund, so if they designate anything, they designate objects (in the broad Fregean sense). I suggest that situations are the things to which gerunds apply.

It's tempting to think of situations as parts of the world. We can do that, as long as we do not think of parthood in a spatial way. Suppose you and I are playing two simultaneous games of chess in our heads (just assume we are that good). It would be very hard to maintain that the two games occupy separate regions of space. Yet, they are distinct situations: it could easily be that with regard to one it is true that white can win in three moves but with regard to the other it isn't. If we are willing to count both of these chess games as parts of the actual world, we are appealing to a notion of parthood according to which the mereological sum of all situations is nothing more or less than the world as it is right now.

[35] Fine (2017) distinguishes three different approaches, which he calls loose, inexact, and exact. These approaches agree on the basic semantic apparatus but disagree on the interpretation of Boolean connectives and quantifiers.

If you are a presentist, you think the world as it is right now *is* the world. If you believe in the past and the future as well, you could say that the world as it is right now is simply the present time. The world as it was five minutes ago is a past time, the world as it will be a year from now is a future time. The world itself is then the sum of all past, present, and future times, which are all sums of past, present, and future situations. If you believe that besides the actual world there are also merely possible worlds, you can think of those too as sums of possible situations. Obviously, if we have merely past, merely future, and merely possible situations and appropriate ways of restricting mereological summation, we have all the worlds and times we need for interpreting modal and temporal operators.

Believing in possible worlds is not the same thing as having a particular take on their nature. Some (actually, very few) think they are concrete particulars on a par with the universe, some think they are properties the universe is apt to instantiate, some think they are states the universe might be in, some think they are pictorial or linguistic representations of the universe, and I am sure there are other options as well. All of these views can be extended with an appropriate mereology and thus accommodate situations. I argued that to adequately account for what we think and say, semantics should countenance situations; I did not say semantics needs to take a firm stance on their nature.

Situation is a broad enough ontological category to model all the other parameters relativists have proposed. Suppose you are convinced that the proposition that roller coasters are fun is true relative to some standards of taste but not others – you can then say that this proposition is true at situations where certain standards of states are at play and false where others are. (What it is for a standard of taste to be at play is a meta-semantic question, not a question for semantics.) Suppose you think that the proposition that the butler might have killed the duchess is true relative to some information but not relative to other – you can say that this proposition is true at situations where that information is available and false where it is not. (What it is for information to be available is, again, a meta-semantic question.) And if you think genuine relativism requires not only the relativity of propositional truth, but relativity of propositional truth to contexts of assessment, all you need to do is to employ in your semantics two contexts – one for utterance and the other for assessment – and let target situations be determined by the latter, not the former. Whether any of this is *needed* to account for our thought and talk is a substantive question much debated in the literature. Here I take no stand on these. What I claim is that debates about various forms of relativism could be seen as debates about what sorts of topic situations there are and how they are to be determined in a conversation.

Situations are also what semanticists following Davidson have been calling events. Originally, Davidson suggested that action verbs have an extra event-argument, that adjuncts are predicates of events, and that action sentences contain an existential quantifier to bind event variables. Ignoring tense and bracketing the semantics of plural definite descriptions, Davidson assigned (9′) as logical form to (9), where the variable was supposed to range over events.[36]

(9) Claire slowly dealt the cards to Dana.

(9′) $\exists e.deal(e, Claire, the\ cards) \wedge slow(e) \wedge to(e, Dana)$

This analysis can account for the validity of an inference form (9) to ‘Claire dealt the cards to Dana’ or to ‘Claire slowly dealt the cards’ (as instances of conjunction elimination within the scope of an existential quantifier) without incorrectly predicting that the inference from the conjunction of these sentences to (9) is valid. What Davidson’s proposal does not predict is the validity of the inference from (9) to ‘Claire dealt slowly’, to ‘Claire dealt to Dana’, to ‘Claire dealt the cards’, and to ‘Claire dealt’. To fix this problem, followers of Davidson suggested treating all the arguments of the verb – with the exception of the event argument – the way adjuncts are normally treated. This can be done if we assume that verbs assign thematic roles to their arguments and if we interpret thematic roles as binary relations between the event the verb describes and the object picked out the by argument. (*Ag* stands for the relation between an agent and an event, *Th* for the relation between a theme and an event.)

(9″) $\exists e.deal(e) \wedge Ag(e, Claire) \wedge Th(e, the\ cards) \wedge slow(e) \wedge to(e, Dana)$

The pattern of inference Davidson sought to account for is quite general – it certainly extends beyond action sentences. But even if ‘Claire lived comfortably in Maine’ entails both ‘Claire lived comfortably’ and ‘Claire lived in Maine’ without being entailed by their conjunction, it still sounds odd to suggest that these sentences quantify over events. What they quantify over are states or processes. Semanticists often call all such entities events while acknowledging that this is an extended use of the term. Events in this technical sense are the sorts of things gerunds apply to – in other words, just the sorts of things I called situations.

[36] Davidson (1967).

Proponents of event-semantics are usually not fans of propositions. Since I am, I would like to raise the question how we should think of the proposition expressed by (9) in light of its proposed logical form (9″). A natural response would be that it is something like the proposition that there is a slow dealing of the cards by Claire to Dana. But this leads to trouble. Suppose Claire dealt multiple times to Dana, sometimes slowly, sometimes not. Suppose (9) is uttered talking about a situation s in which Claire's dealing was not slow. We would like to say that the proposition expressed by (9) is false (or, at least, not true) at s. But it's hard to see why it would be untrue at s that there is *some* slow dealing of the cards by Claire to Dana, as long as we do not require that this be s itself. So, we should ditch the existential quantifier and bind the situation variable by a lambda abstractor.[37]

(9‴) $\lambda s.\mathit{deal}(s) \wedge \mathit{Ag}(s, \mathit{Claire}) \wedge \mathit{Th}(s, \mathit{the\ cards}) \wedge \mathit{slow}(s) \wedge \mathit{to}(s, \mathit{Dana})$

If declarative sentences are predicates of events, events are situations, and declarative sentences express propositions, then propositions are properties of situations. Truth-at thus turns out to be truth-of: when an act of assertion or a state of belief is true *simpliciter* that is because the proposition it expresses is truly predicated of the situation the assertion or belief is about.

The inferences that motivated the Davidsonean semantics can still be accounted for, assuming we employ a conception of validity that is apt for dyadic truth. Let's say that an inference is valid iff whenever each premise is true of some situation the conclusion is also true of that situation. Then, the inference from 'Claire slowly dealt the cards to Dana' to 'Claire dealt the cards to Dana' and 'Claire slowly dealt the cards' is valid – if a situation is a slow dealing of the cards by Claire to Dana then it is also a dealing of the cards by Claire to Dana and a slow dealing of the cards by Claire. But the inference from 'Claire dealt the cards to Dana' and 'Claire slowly dealt the cards' to 'Claire slowly dealt the cards to Dana' is not valid – there could be a situation where Claire dealt the cards to Dana and another where she slowly dealt the cards without there being a situation where she slowly dealt the cards to Dana.

[37] Similar logical forms are quite standard in semantics since Berman (1987) and Kratzer (1989), although typically they involve extra complexity. For a fairly detailed compositional semantics, see Elbourne (2005). I stress that the ontological assumptions these authors embrace as well as many of the semantic details are negotiable.

To say that propositions are properties may sound like a category-mistake – I propose to think of it as a substantive analysis. Lewis initially characterized propositions as properties of possible worlds, but in light of *de se* attitude ascriptions he ultimately settled with the broader view, according to which they are properties of possible objects.[38] Since situations are objects, this proposal is in the same ballpark as my own view. Lewis was also committed to another reductive analysis: the claim that properties are just sets. I say no such thing. I hold open the possibility that to provide an adequate account of mental state and speech act ascriptions we must ultimately individuate properties of situations more finely than the set of their possible instances.

One might complain that it is unnatural to assign the same kind of semantic value to full sentences and bare verbs, adverbs or prepositional phrases. In my view these are all propositions. How could 'deal' and 'Claire dealt the cards to Dana' have the same kind of meaning? There are two reasons one may feel this to be unnatural; to my mind, neither is particularly persuasive. One reason for insisting on a special semantic value for sentences is their connection with illocutionary force. It is sometimes suggested that we cannot make assertions uttering sub-sentential expressions. But, apparently, we *can*: we can hold up a letter and utter 'From Spain' thereby asserting that the letter is from Spain.[39] The other reason for wanting a special semantic value for sentences is their particular pattern of syntactic distribution. Sentences can certainly not be substituted *salva beneformatione* for mere verbs and verb phrases. Still, we *do* assign the same type of semantic value to lots of expressions whose syntactic distribution is wildly different: the complements of 'believe' and 'want' are supposed by nearly everyone to be both propositions, yet most complements of one cannot be substituted for complements of the other. In the end, what is special about sentences is that they are *syntactically complete* – they contain a verb and all the obligatory thematic arguments lexically associated with expressions within them are saturated. It is not clear that syntactic completeness is a mark of a distinctive kind of meaning.

[38] Lewis (1979).

[39] For detailed arguments, see Stainton (2006).

7. Closing

In a recent article, Scott Soames presented the following argument in favor of the thesis that fundamental truth is monadic:[40]

> "For a sentence *S* (which is used to make assertions and express beliefs) to have a meaning, or semantic content, is for *S* to express a proposition that represents something as being some way or other. In virtue of this, we speak derivatively of *S* representing things. 'Snow is white' represents snow as white, while 'The U.S. President is male' represents the property being U.S. President as uniquely instantiated, and being male as instantiated by whatever instantiates being U.S. President. A meaningful sentence of this sort represents the universe (or parts of it) as being a certain way (or ways). Its truth conditions follow from this; if *S* (simply) represents A as being B (and nothing else), then *S* is true iff A is B. We have no idea what it is to be representational, and hence meaningful, apart from having such (monadic) truth conditions."

I agree that for a meaningful sentence to represent, it must represent something as being a certain way, and that if *S* (simply) represents A as being B (and nothing else), then *S* is true iff A is B. What I disagree with is the way Soames identifies the A and B in the particular cases he mentions. 'Snow is white' does not represent *snow* as being white – it represents, in use, a *situation* (perhaps as large as the whole world) as being one where snow is white. And 'The U.S. President is male' does not represent a *property* as being instantiated in any way – it represents, in use, a *situation* as being one where the U.S. president is male.

The central point of contention is, I think, the one Soames touches upon at the very beginning of the argument. He says propositions represent things as being some way. I say propositions are ways things can be represented as being. Fundamental representations – the things that represent and without which nothing would represent – are mental states. Some of these have propositional content. The ones that do represent not only in virtue of having that content, but also in virtue of predicating that content of situations they are about. As Austin put it: "It takes two to make truth".[41]

[40] Soames (2010, 125).

[41] Austin (1950/1961, 124, n. 1).

Acknowledgements

I thank audiences at the 8th GAP Conference at Konstanz, at NYU, and at Lewis and Clark for comments, objections, and general discussion. Special thanks to David Chalmers, Troy Cross, Matti Eklund, Kit Fine, Tamar Szabó Gendler, Zoltán Molnár, Jim Pryor, Jonathan Schaffer, and Bruno Whittle for making me rethink some of my arguments.

References

Austin, J. L. 1950/1961. Truth. In J. D. Urmson and G. Warnock (eds.) J. L. Austin: Philosophical papers. Oxford: Oxford University Press. 117–133. Originally appeared in J. L. Austin 1950, Truth, Proceedings of the Aristotelian Society, Supplementary Volume 24, 111–128.

Bach, K. 2001. You don't say. Synthese 128. 15–44.

Barwise, J. 1989. Situations, facts, and true propositions. In J. Barwise, The situation in logic. Stanford: CSLI Publications. 221–254.

Barwise, J. and J. Etchemendy. 1987. The liar: An essay on truth and circularity. Oxford: Oxford University Press.

Berman, S. 1987. Situation-based semantics for adverbs of quantification. In J. Blevins and A. Vainikka (eds.) Studies in semantics. Amherst: University of Massachusetts. 46–68.

Cappelen, H. and J. Hawthorne. 2009. Relativism and monadic truth. Oxford: Oxford University Press.

Cappelen, H. and E. Lepore. 2005. Insensitive semantics. Malden, MA & Oxford: Blackwell.

Carston, R. 2002. Thoughts and utterances. The pragmatics of explicit communication. Oxford & Cambridge, MA: Blackwell.

Davidson, D. 1967. The logical form of action sentences. In N. Rescher (ed.) The logic of decision and action. Pittsburgh: Pittsburgh University Press, 81–95. Reprinted in D. Davidson 2001, Essays on actions and events, Oxford: Clarendon Press.

Elbourne, P. D. 2005. Situations and individuals. Cambridge, MA: MIT Press.

Evans, G. 1985. Does tense logic rest on a mistake? In J. McDowell (ed.) G. Evans, Collected papers. Oxford: Clarendon Press. 343–363.

Fine, K. 2017. Truthmaker semantics. In B. Hale, C. Wright and A. Miller (eds.) A companion to the philosophy of language. Malden, MA & Oxford: Wiley-Blackwell. 556–577.

Geach, P. 1956. Good and evil. Analysis 17. 32–42. Reprinted in P. Foot (ed.) 1967, Theories of ethics, Oxford: Oxford University Press, 64–74.

Heim, I. and A. Kratzer. 1998. Semantics in generative grammar. Malden, MA & Oxford: Blackwell.

Jacobson, P. 1999. Towards variable-free semantics. Linguistics and Philosophy 22. 117–184.

Kaplan, D. 1989. Demonstratives. An essay on the semantics, logic, metaphysics, and epistemology of demonstratives and other indexicals. In J. Almog, J. Perry and H. Wettstein (eds.) Themes from Kaplan. Oxford: Oxford University Press. 481–563.

Keshet, E. 2008. Good intensions: Paving two roads to a theory of the *de re/de dicto* distinction. Doctoral dissertation. Massachusetts Institute of Technology.

Kölbel, M. 2002. Truth without objectivity. London: Routledge.

Kratzer, A. 1989. An investigation of the lumps of thought. Linguistics and Philosophy 12. 607–653.

Kratzer, A. 2009. Making a pronoun: Fake indexicals as windows into the properties of pronouns. Linguistic Inquiry 48. 187–237.

Lewis, D. 1979. Attitudes *de dicto* and *de se*. The Philosophical Review 88. 513–543. Reprinted in D. Lewis 1983, Philosophical Papers, vol. 1, Oxford: Oxford University Press, 133–155.

Lewis, D. 1980. Index, context, and content. In S. Kanger and S. Ohman (eds.) Philosophy and grammar. Dordrecht: D. Reidel. 79–100. Reprinted in D. Lewis, Papers in philosophical logic, Cambridge: Cambridge University Press, 1989, 21–44.

MacFarlane, J. 2005. Making sense of relative truth. Proceedings of the Aristotelian Society 105. 305–323.

Moore, G. E. 1903. Principia ethica. Cambridge: Cambridge University Press.

Percus, O. 2000. Constraints on some other variables in syntax. Natural Language Semantics 8. 173–229.

Recanati, F. 2002. Unarticulated constituents. Linguistics and Philosophy 25. 299–345.

Recanati, F. 2004. Literal meaning. Cambridge: Cambridge University Press.

Recanati, F. 2008. Moderate relativism. In M. García-Carpentiero and M. Kölbel (eds.) Relative truth. Oxford: Oxford University Press. 41–62.

Recanati, F. 2010. Truth-conditional pragmatics. Oxford: Clarendon Press.

Soames, S. 2010. True at. Analysis Reviews 71. 124–133.

Stainton, R. 2006. Words and thoughts. Oxford: Oxford University Press.

Stalnaker, R. 1998. On the representation of context. Journal of Logic, Language and Information 7. 3–19.

Stanley, J. 2000. Context and logical form. Linguistics and Philosophy 23. 391–434.

Stanley, J. 2002. Making it articulated. Mind and Language 17. 149–168.

Stanley, J. and Z. G. Szabó. 2000. On quantifier domain restriction. Mind and Language 15. 219–261.

Szabó, Z. G. 2000a. Adjectives in context. In I. Kenesei and R. M. Harnish (eds.) Perspectives on semantics, pragmatics, and discourse. Amsterdam & Philadelphia: John Benjamins. 119–146.

Szabó, Z. G. 2000b. Descriptions and uniqueness. Philosophical Studies 101. 29–57.

Szabó, Z. G. 2005. The loss of uniqueness. Mind 114. 1185–1222.

Szabó, Z. G. 2010. The determination of content. Philosophical Studies 148. 253–272.

Szabó, Z. G. 2019. Semantic explanations. In E. Lepore and D. Sosa (eds.) Oxford studies in philosophy of language. Oxford: Oxford University Press. 240–275.

Szabolcsi, A. 1987. Bound variables in syntax: Are there any? In L. Groenendijk (ed.) Proceedings of the 6th Amsterdam Colloquium. Amsterdam: ITLI Publications. 331–353.

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 529–540

Interpretation of the concept of namespace

ISTVÁN SZAKADÁT
Budapest University of Technology and Economics
i@syi.hu

KEYWORDS

interoperability
name space
proper name space
Knowledge Organization System
semantic structure

ABSTRACT

The significance of namespaces is becoming more and more recognized with the spread of the world of networks and digital culture. Global, national and local namespaces are being built around the world. Before clarifying what the concept of namespace means (and what it does not), I will present some special namespaces in order to show how they work and what their function is. After that, I will interpret the concept of namespace and then briefly examine the practical implications of the theoretical findings.

1. File names and the namespace

Perhaps the simplest but surely the most commonly used namespace is the computer file system, whose only task is to ensure that each file has a unique name on the computer. It depends on the operating system what rules can be used to create names, but this is a secondary issue as far as the functioning of the namespace is concerned. However, the creation of unique names is facilitated by the possibility to build names from components. Files can be collected into groups (folders, directories), folders can be embedded into each other, and the complete (unique) file name can be obtained by concatenating the (short) name of the file and the names of the folders containing the file. This is a hierarchical method of name-identification, with all the advantages and disadvantages of hierarchical control. From a linguistic point of view, hierarchical organization means that names can be interpreted as compositions of distinct name elements, so a name can be used more than once as a name element when composing a specific file name. This is made possible by the fact that file names are compound names, i.e. the full and complete (thus unique) file names contain at least as many name components as the number of folders the file itself is embedded in, starting from the root of the file system. However, these name elements do not have to be unique in themselves, they can be duplicated.

With the requirement that each file should have a name and this name should be unique, it can be ensured that each file is identified unambiguously, and always found on the computer. This expectation is compatible with a file having multiple names in the file system, since if the uniqueness of names is guaranteed and all names are linked to files, then all files can be found on the basis of the names. (It would not be true the other way round, but it seems to be pointless to find a name starting from a file.) In linguistic terms, we can say that if two names belong to the same thing (here: file), synonymy is allowed in the namespace, but if a name can only belong to one thing (file), homonymy is not allowed.

It is also important to note that the file system as a namespace is local in the sense that there is at least one file system on each computer, so there are at least as many namespaces as computers. Of course, these local namespaces can be combined (by connecting them to a network) and therefore larger namespaces can be created. Their uniqueness is easy to maintain if the machines connected have unique identifiers. With respect to a file system, it is easy to answer the following question (which is important for each namespace): Who can manage (create, delete, change) the names? To simplify things a little, the answer is that it is the owner (user) of the computer.

2. Domain names and the namespace

Similarly to the case of the file system, the expectation for the uniqueness of names and for the accessibility of the computer resources denoted by names arises with respect to the domain names of the World Wide Web, too. The system of domain names is a namespace as well. Both the domain names on the web and the hierarchically structured set of web page addresses can be regarded namespaces (the latter containing the former). What is the main function of domain names? To mark the common roots of web page addresses (facebook.com, microsoft.com, mit.edu) with character series that can be unambiguously identified and easily memorized by humans. There are rules for creating domain names (concerning what characters can and cannot be used as name elements, or stating the fact that separators have to be used among the name elements, etc.) and there is a basic expectation for the whole set of domain names: each name must be unique. This can be achieved by distributing the task of registering domain names among specific institutions, which are expected to comply with the principle of unique name distribution in the namespace of their responsibility. This naming system is based on hierarchical organization, too.

Maintaining the domain namespace is necessary in the first place so that, in the case of a new naming request, it would be possible to know whether the required domain name exists in the namespace. If it is a name already present in the namespace, it cannot be registered. If the required name is not yet in the namespace, it can be registered (provided it complies with the naming rules). What does this registration mean? It entails establishing a connection between domain names and the owners of domain names. The other important function of the domain namespace is to create a clear link between domain names and IP addresses that belong to computers.

Domain namespace is thus a hierarchical registry of unique names and the dynamic connections between names and owners on the one hand and names and physical addresses on the other. The uniqueness of the domain names ensures the unambiguous identification of their IP addresses and the registration provides the right of use of the domain name for its owner. As the main function of the domain namespace is the clear identification of all communication nodes on the web, it is obvious that this is a global namespace. This namespace allows synonymy but forbids homonymy, and uses the technique of creating composite names, since domain names can be divided into components, and the same name elements may be reused in other domain names as well.

3. Person names and the namespace

After the two examples from the world of technology, let us look at why and how person names are used and how to build namespaces for person names. In a family, the use of different first names obviously has the same meaning as in the two examples above: by using unique names, we create the possibility of referring unambiguously to the entities denoted by the names, the children within the family. For a family, it is enough to choose names from a collection of a few thousand first names that have become acceptable in the course of time, but beyond family communities, such a quantity of names is clearly unsuitable for the identification of every child. From this point of view, the formation of compound names where the full name of the person consists of the concatenation of a family name and a first name can be a bit of help, but we know that this solution cannot ensure that everyone has a unique name. With the help of composite person names, we can only identify people clearly if we determine the domain of people, in other words, we create a naming context with a precise boundary. What does this mean? In narrow contexts (family, small community, clan, village) there could be hope that all people can be identified with the help of family names and first names, but in a wider context this hope is eliminated because homonymy will become more and more common.[1]

The name *John Smith* is no longer able to identify a person unambiguously if there are several families in the village called *Smith* that have given the first name *John* to one of their sons. This example highlights the importance of the naming context. Within the family as a naming context, the good functioning of the namespace can be guaranteed; everyone can have a unique name. If this context is expanded, the homonymy-free state can still be maintained for a while, but expansion can reach a limit where it cannot be assured that one particular name denotes only one person. At this point, the namespace becomes useless or, more precisely, partially useless, as it may have some segments that can still function well, but the functionality of the namespace can only be restored if the scope of the naming context is narrowed to the required size.

The example of the person namespace indicates that names in themselves are not able to identify the entities denoted, only with the use of the context and the names together can we hope for the successful operation of namespaces.

4. The components of the namespace

The next important question is what the components of namespaces are. It is obvious that whatever one means by namespaces, names are always among their elements. On its narrowest possible interpretation, a namespace could be defined as a simple collection or a list of names, where the type of objects they denote is also specified. According to this approach, a namespace is a unique list of names that can be used for some purpose. This means that in a geographical namespace the name *Old Hill* could only appear once, and the only thing we could say about it is that it possibly points to one or more geographical locations. Similarly, a family namespace interpreted in the narrowest sense could only be expected to include all possible family names in a manner that every name only occurs once. Such

[1] Synonymy does not cause any problems here, either. One person may have several names; if they are unique, each of them may be suitable for identifying the given person.

a namespace must be disjoint and exhausted simultaneously (Bittner et al. 2004). Disjointness ensures the uniqueness of the names, and exhaustedness guarantees that all possible usable names are included. A namespace defined so narrowly could be used to control the process of selecting names in a data system by requiring that at certain input points only elements of a given namespace (family namespace) can be chosen. Such a namespace, however, would not have too many practical benefits, it would not help to reach the goal of uniquely identifying the set of things to be described with namespace elements. For such a purpose, a namespace like the one above would only be suitable if the number of names were larger than that of the things to be denoted, or if the names could be reused with the help of a hierarchical name composing technique. For people, organizations, geographical locations and many other things that could not be guaranteed, so in such contexts no homonymy-free state can be achieved.

Returning to the question of what the additional elements of the namespace can be, the answer must take into account the purpose for which the namespaces are used. If we want to use namespace elements to identify things of a certain type, then we have to include in the namespace the entities that the names refer to. These can be called *name holders*. In afile system files are the name holders that can be identified by file names, and in a domain name system IP addresses (and content packages packed there) are the name holders that can be identified by domain names. A name holder may be a person who can be referred to with one or more names but name holders may be organizations or geographical locations, too.

If namespaces contain not only names but also name holders, they must include something else, too: *the relations between names and name holders*. In fact, managing these relations is the real sense of the use of namespaces.

5. The definition of the namespace

Namespaces can be defined on the basis of the quality of the system of relations between names and name holders (dynamically changing in space and time): a *namespace is a function that returns one and only one name holder for each possible naming*. This definition seems simple, but the question is how a namespace defined in the manner above can be used in practice.

In the case of person namespaces it has already turned out that the names themselves are not capable of unambiguously identifying people (name holders): several people may be named *John Smith*. The question is how the namespace can operate as a function under these conditions. In order to be able to proceed, the concepts we have used so far need to be be clarified further. The first clarification can already be seen in the definition of the namespace above, which does not contain the term *possible name* but *possible naming*. What is the difference between these two terms and why is it necessary to introduce the new category (naming)?

It is true for all of the namespaces referred to so far (even if in some cases it may be surprising) that the names that appear as elements of namespaces can be interpreted as *proper names*. For person names, names of geographical locations, organisations this is self-explanatory, it is easy to see and accept it for domain names as well, but in the case of file systems this qualification may be surprising for the first time. But it is true. File names play the same role for files as person names do for people or names of geographical locations do for geographical locations. They refer to a specific existing entity (file, person, geographical location) in order to distinguish this particular thing from the other similar, specific things (the given file from the other files, the given person from the other people, etc.). When using names for this purpose, we use special names: proper names.

A unique feature of proper names, which differentiates them from common names, is that proper names are rigid designators (Kripke 1980). Philosophers are in dispute about whether proper names have a meaning but there is a consensus among them concerning the fact that with a proper name we refer to a single individual, and this reference is rigid in the sense that under all possible conditions (in all possible worlds) the reference remains permanent between the proper name and the individual referred to. This is not the case with common names, where the scope and extension of names may change in different conditions (in different possible worlds). However, the question arises as to how names (especially proper names) can fulfil their function if they are very different from the denoted entities in an important philosophical quality. What are we talking about?

When talking about name holders and names that we use for identification, we always talk about specific individuals (people, group of people, institutions, geographical locations places, books, IP addresses, etc.) that can be placed in some kind of category. Their existence, concreteness and uniqueness cannot be disputed: they are all unique in the sense that they can be always localized in space and time.[2] However, when we refer to these

[2] A defining quality of individuals is that they are always connected to space and time. In the case of an individual we must always be able to determine (in principle) where he/she “is located” in space and time. A building, a mountain, a person, a book, a video cassette, a CD can always be clearly localized in spatial and temporal dimen-

individual name holders with the help of their names registered in a given namespace, we can face a serious problem because of the time-and-space-independent quality of the names. Names can exist in more than one place at the same time. As far as a person is concerned, we can tell a exactly where he/she is at a given time, but we cannot say that “his/her name is here now”. This is exactly the source of the phenomenon of homonymy. When we want to identify a person with a name, we might face the same problem: the same name can be assigned to several people. The big question is then the following: how can namespaces be used for identification if proper names themselves are not suitable for that purpose?

6. Unambiguous names

We can ask the question a little bit differently: how can (proper) names be made unique in such a way that they become fit for identifying individual name holders? The solution might be that for the identification we use *namings* (naming events or naming states) and not proper names. This solution can be traced back to Kripke’s interpretation of proper names (Kripke 1980). According to Kripke, a reference of a proper name can be determined not by any description associated with it but by the naming events associated with the name in a historical chain:

> “[...] when I pronounce the name ‘Ernő Rubik’, its reference is the person who is determined by the historical chain (or rather web) of the use of names associated with my utterance. At the beginning of the chain there is the introductory use of the name, which is followed by the forwarding (repeating) use of the name, by which the name ultimately came to me.” (Zvolenszky 2015)

Giving and using a proper name means creating and maintaining a rigid designator in the practice of a naming community. The new elements here are the concepts of naming and of naming practice. By means of “pointing” and naming gestures in a community’s practice, a name gets attached to

sions. Geographical entities are to be determined in the first place by specifying where their components are located in a geographical area at a given time. In principle, we could identify any person by specifying the segment of the space at a particular time where he/ she is, but this would not be economical in terms of its informational needs, since we should know about each and every one their temporal and spatial locations at all times. It is more practical to identify remarkable events from people’s lives and to link them to space and time coordinates. Such a notable event is the birth and death of a person, which we can “catch” with the space and time coordinates of the two events.

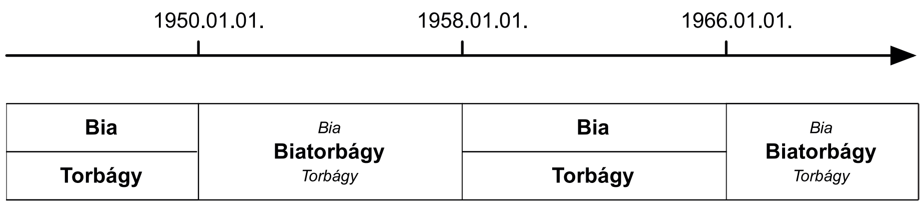
a name holder and this maintains a permanent relation between the name and the entity referred to through a coherent network of naming events. It is not the (proper) name itself that identifies but the naming practice of the naming community, the naming chain, which in turn means a chain of unique events. This is important because individual name holders can be identified through individual naming events. This will eliminate the philosophical difference between names and name holders, since a naming as event and a name holder are both individual entities. Of course, the same proper name can be applied to different name holders through different naming events, simultaneously or at different times. This interpretation can thus handle the phenomenon of homonymy that we mentioned earlier. Naming is not merely a one-shot action but a continuous naming sequence (or just a naming state). This solution is able to ensure the identification of specific names because a naming (as an event or state) can already be tied to space and time coordinates. This spatial and temporal localization of the naming event can be interpreted as a way of creating the context for the given proper name that allows the latter to become appropriate for unique identification.

We have not yet mentioned the fact, but need to do it now, that names and naming events can belong to different types, so we can typify both names and naming events. As to person names, we can talk about nicknames or pseudonyms, official names, birth names, pen names, etc. These may all relate to different naming practices and communities, and of course it may be the case that several name using communities have the same name for someone. Viewing the naming practice as a context can also handle the phenomenon of two names being identical in two naming practices but still belonging to different name types within the two contexts.

After this long introduction, let us turn to the question of what follows from the theoretical considerations for the concrete practice of building namespaces. They have several important consequences.

Due to the fact that the namespace has two equally important components, the name and the name holder (the one that refers and the one referred to), both components must be handled in the namespace. In practice, this means that both name holders and names must be identified in some way, and (probably most importantly) there exists a third component, the relationship between them, and it should also be included in the namespace. The reason why this is important is because the phenomenon of synonymy is very common in the case of all types of proper names (person names, company names, geographical names, etc.), that is, a name holder may easily have several proper names. This should be treated as follows: first, we identify the name holder and all the names associated with it in themselves, and, second, we record all the naming events as many times as the naming event (praxis) occurs in the case of the given name holder. Below we show why this is necessary.

It is enough to clearly identify the naming event as a context, there is no need for a deeper analysis. In a person name space, by connectingthe unique identifier of the name holder (*personid*) and the unique identifier of the associated name (*nameid*) in a manner that identifies the type of the connection as well we can capture the naming context itself. Let us look at the example of Biatorbágy (a Hungarian settlement). The independent villages Bia and Torbágy were united in 1950 and the name of the new settlement between 1950 and 1958 was Biatorbágy. In 1958 the two villages separated from each other again and officially got back the names Bia and Torbágy. Finally, they were reunited in 1966, and from then on the town is named Biatorbágy again. The two formerly autonomous settlements, Bia and Torbágy, were officially named by their original names twice, and the name of the united settlement was Biatorbágy in two periods. We thus have three names but six naming events (name usage practices). At the level of names, the names of Bia, Torbágy, and Biatorbágy have existed since their creation, they never ceased to exist, but at the level of naming events or naming practices they did not always exist. This example also shows that when we connect a name and a name holder in a naming event, it could happen that we build the same link between the same name and the same name holder in different times, but these connections can be instantiated and identified by different naming events.



Naturally, it is the names that are of primary importance for people and for them the multiple uses of the names (as it was shown in the case of Bia, Torbágy, Biatorbágy) do not need to be separated, but for a precise, machine based and scientific processing, these differences need to be handled.

In namespaces, we associate proper names with name holders through naming events. So far we have talked about namespaces as having only proper names. After analyzing issues related to the management of proper names, we should also look at the issue of common names. There are language systems that contain common names and it can be useful to clarify their relation to namespaces.

We have already distinguished between the narrower and broader interpretations of namespaces. In the narrower sense, by the concept of namespace we mean a unique list of proper names, which can be used in the namespaces defined in the broader sense. We can also compile such collections from common names in order to characterize a certain domain of knowledge by giving the list of valid, useable common nouns (concepts) and the relations between these names (concepts). The latter is, however, a different kind of task and possibly yields different results. While proper names can only be organized in a semantically flat structure, common names can be formed into a semantically rich and complex structure. Such a structure of concepts can be used to represent our knowledge about the world.

7. Name structures - Knowledge Management Systems

A structured collection of names, i.e. a set of names and a set of relations defined on them is called a mathematical *structure*. In other knowledge domains, the concept of *Knowledge Organization System (KOS)* is used for the same phenomenon. We can distinguish a variety of knowledge organization systems, proceeding from simpler structures to more complex ones: the *term list*, the *classification system*, the *thesaurus* and the *ontology.* We can also differentiate among the latter according to the logic of their construction, and if we take into account the problem of control, we can distinguish between controlled and non-controlled KOS. Non-controlled KOS's are exemplified by the so-called *folksonomy.*[3]

Term lists simply list the names related in some sense, and apart from the lexicographic ordering, other relations cannot be defined on the elements of the set.[4] The elements of classification systems are linked by a subordination (containment) relation, resulting in a hierarchical structure of names. Since in many cases a single hierarchical relation is used instead of several, semantically different subordination relations when constructing classification systems, it is possible that a semantically inconsistent structure is created. The "weakness" of the classification systems is that only

[3] The term list, the classification, the thesaurus, and the ontology are controlled KOS.

[4] Proper name spaces with a narrower meaning can also be qualified as term lists.

one subordination relation is used to express both generic (subclass/class), partitive (part/whole), and even instance-of relations. This vagueness is eliminated by the thesaurus, which defines multiple relations (generic and partitive subordinate-of, instance-of, synonym-of, etc.). This creates a more complex structure than the ones before. Ontologies move further towards deep structuring in the sense that they allow the introduction of any arbitrary, formally defined relation, which can create an even more complex structure than a thesaurus. The knowledge organization systems outlined here are to be considered controlled systems in the sense that both the elements of the systems and the relations between them can only be introduced into the system by authorised people. Folksonomies are different in this respect because they are built without any control. Folksonomies do not handle relations between elements, but any person who uses them for something may add elements. The applicability and desirability of folksonomies is greatly enhanced by the possibility that tags generated by the users can be connected to the traffic (usage) data resulting from user activities, which can be used as a quality assurrance filter.

Knowledge organization systems collect the terms needed to describe the world, and through the relations between the terms they facilitate access to the terms themselves. The different kinds of knowledge organization systems differ from each other in their relations only, and theoretically there is no ontological constraint on the usable terms. In principle, each system is capable of covering any knowledge domain (ontology) with its terms.

Knowledge organization systems are, in a sense, insensitive to whether they have to handle proper or common names; it often happens that a particular knowledge organization system contains both proper and common names.

The concept of namespace can be applied to knowledge organization systems, which appears understandable and manageable on the basis of the considerations above. It would be reasonable to distinguish the proper name spaces and common name spaces more from each other but it is more important to know about a given system what types of names it is built up from. If relevant, proper name spaces can be marked with an appropriate qualifier.

After this sketchy review of knowledge organization systems, we have to answer the question of what such systems can be used for. They are most often used to provide a suitable set of terminology for a knowledge domain – to support content description and to facilitate document retrieval.

8. Name structures – Document Descriptor Systems

Both namespaces and knowledge management systems can be interpreted as structures whose semantics depends partly on the relations applied in the system and partly on the ontological commitments made in the course of the compilation of the term set. These systems can be applied to all knowledge domains, there is no area in which they could not be used.

However, there are other knowledge management systems that aim to deal with knowledge from a given knowledge area only, based on a specific ontological commitment. To narrow our focus to the world of libraries, we can mention the Marc21, RDA or BIBFRAME systems. They also have semantics, they can also be considered complex structures, but their functionality is different from the KOS systems discussed earlier. They can use proper and common name spaces, or knowledge organization systems, but their main purpose is more specific: the description of a certain type of documents. They can achieve this goal by outlining a scheme representing the ontological commitment and the knowledge of the given profession, in which everything can be said about the type of document under consideration. This scheme contains all the entities that are needed for a professional description of the given knowledge domain, as well as the relations between the entities and the points where "external" namespaces and knowledge organization systems can be made use of. These document descriptor systems may be connected to namespaces but their purpose and content is different.

The real significance of namespaces is the unambiguous identification of names in the ontological segment that they want to describe. By completing this task, the inventory systems of libraries, museums, archives, etc., which associate the identifiers of namespaces with their own identifiers can be made interoperable because they are able to refer to certain entities of the world in the same way due to the namespaces they all use.

References

Bittner, T., M. Donnelly and B. Smith. 2004. Individuals, universals, collections: On the foundational relations of ontology. In A. C. Varzi and L. Vieu (eds.) Formal ontology in information systems. Proceedings of the Third International Conference (FOIS 2004). Amsterdam: IOS Press. 37–48.

Kripke, S. A. 1980. Naming and necessity. Cambridge, MA & Oxford: Blackwell.

Zvolenszky, Zs. 2015. Pofonegyszerű, kedves Kripkém? (Saul Kripke: *Reference and existence*. Oxford University Press, 2013). Műút 2015. (http://www.muut.hu/archivum/15361)

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 541–559

Forgotten islands of regularity in phonology

ANSSI YLI-JYRÄ
anssi.yli-jyra@helsinki.fi
University of Helsinki

KEYWORDS

Generative Phonology
finite-state phonology
regularity
linear-time Turing machines

ABSTRACT

Hennie presented a very general sufficient condition for regularity of Turing machines. This happened chronologically before Generative Phonology (Chomsky & Halle 1968) and the related finite-state research (Johnson 1972; Kaplan & Kay 1994). Hennie's condition lets us (1) construct a finite-state transducer from any grammar implemented by a linear-time Turing machine, and (2) to model the regularity in context-sensitive derivations. For example, the suffixation in hunspell dictionaries (Németh et al. 2004) corresponds to time-bounded two-way computations performed by a Hennie machine. Furthermore, it challenges us to look for new forgotten islands of regularity where Hennie's condition does not necessarily hold.

1. Introduction

Generative Phonology (Chomsky & Halle 1968) is a rule-based string rewriting system that has been scrutinized carefully over the years of its existence. One of the major weaknesses of the system is that it has been proven to be equivalent to Turing machines (TMs) (Chomsky 1963; Johnson 1972; Ristad 1990). As the derivations of such a machine do not necessarily terminate, the system is seriously defective and impossible to falsify. Thus, an unrestricted rewriting allowed by Generative Phonology does not make a very good scientific theory in the light of Popper (1959), see also Johnson (1972, 32).

In spite of the original shortcomings and the increased depth in the current phonological theory, the original "SPE" formalism is interesting for its own sake. First, the formalism has been employed extensively in natural language processing and descriptive linguistics. There, it has been used to express phonological generalizations based on empirical data. Second, revisiting the original formalism and its decidable subsets can produce valuable ideas that are applicable to more ambitious theories, such as Optimality Theory (Prince & Smolensky 2004) and Harmonic Serialism (McCarthy 2000).

1.1. The well-known islands of regularity

When realistic grammar instances in Generative Phonology have been studied closely, a striking contrast between the original undecidable theory and the actual grammars has been discovered. In pioneering studies (Johnson 1972; Kaplan & Kay 1994), most practical grammars in Generative Phonology have been shown to *satisfy* a two-part condition under which they correspond to finite-state transducers:

1. *Non-self-embedding.* Directional or simultaneous context-sensitive rules whose non-contextual parts do not apply to their own output are finite-state (Kaplan & Kay 1994, 363, 365).
2. *Finite composition.* If a grammar is defined as a finite sequence of rewriting rules, each of which is a regular relation, then the grammar as a whole represents the regular relation given by their composition (*ibid.*, 364).

These observations have led to the development of algorithms for transforming restricted fragments, or *islands*, of Generative Grammar into finite-state transducers. For example, the algorithm of Mohri & Sproat (1996) constructs transducers from rules that are applied in a directed fashion. Karttunen (1995) treats various application modes and different types of context conditions. These finite-state islands in Generative Phonology have become standard textbook material (Jurafsky & Martin 2000; Beesley & Karttunen 2003), and many redesigned compilation algorithms have been proposed to pursue efficiency, flexibility and the generally correct semantics.

The literature of methods that compile individual rules into finite-state transducers suggests that regularity of phonological grammars is to be proven inductively, by using operations that preserve regularity of regular relations. But we should not overlook a more extensive picture of regularity as a property of the relation rather than as a property of the construction. Therefore, we should now start to pursue for a wider understanding of the *archipelago* of finite-state islands in Generative theories as well as in all computational models of language.

1.2. The search for further islands

Proving that the input-output relation defined by a grammar is regular is a complicated task. The known finite-state islands and the closure properties of finite-state transducers solve only the easy cases where the application order of rules is fixed and the rules can be combined under a finite composition. But if the grammar contains iterative rules, we do not have a general method that would return a non-iterative grammar. The regularity of the string relation defined by iterative rules is computationally undecidable already for context-free grammars (Stearns 1967; Greibach 1968), not to talk about Turing machines and equivalent grammar systems.

In light of this, we see that the fundamental results in finite-state Phonology (Johnson 1972; Kaplan & Kay 1994) have given us only islands, sufficient conditions where the grammars or parts of grammars are finite-state and generate regular relations. They do not exclude new conditions that can also be valuable. New conditions are, ideally, constructive and turn a formerly nonconstructive property into a method that gives a finite-state transducer.

For example, it has been obvious since Chomsky & Halle (1968) that a phonological grammar is regular when it contains only right-linear (or left-linear) rules (Chomsky 1963). The left-linear rules have the general shape $\alpha \to \beta\gamma$ where α, β, γ are symbols and β does not match any left-hand side in the grammar rules. The achievement of Johnson (1972) was to expand the default regular subset of Generative Phonology by showing that the linear and the simultaneous application of phonological rules with context conditions can also generate a regular relation.

Kaplan and Kay (1994) also discuss general situations that are usually known as cyclic derivations (Mohanan 1986). For example, the word *unenforceable* has the recursive structure [*un*[[*en*[*force*]]*able*]]. Here the phonological rules are applied first to the innermost part, *force*. Then the innermost brackets are removed and the application is repeated until no brackets are left. Kaplan and Kay point out that “there may be restrictions on the mode of reapplication that limit the formal power of the [cyclic] grammar...”. However, Kaplan and Kay (1994, 365) seem to think that these restrictions are analogous to context-free grammars with only right- or left-linear rules.

Besides context-free grammars with only right- or left-linear rules, there are also self-embedding grammars that generate regular languages. For example, the context-free grammar $S \to aS \mid Tb; \quad T \to Tb \mid c$ generates the regular language a^*cb^* and the context-sensitive grammar

$S \to aS$; $aS \to abT$; $bT \to cbT \mid c$ generates the regular language a^*c. In these examples, the grammars look simple but are not immediately regular on the basis of the shape of their rules.

In the sequel, section 2 presents a concrete example of a very simple context-sensitive formalism whose conversion to a finite-state equivalent grammar is tricky. In section 3, the reader is familiarized with a one-tape Turing machine and Hennie's sufficient condition for regularity. The paper closes with remarks in section 4.

2. Safe unbounded composition

We will now give an example of a grammar whose regularity is not obvious on the basis of the standard conditions.

2.1. hunspell

Our example of a non-classical finite-state grammar is the `hunspell` formalism (Németh et al. 2004) that represents a stage in the development of spell checking algorithms. We only discuss its suffix rules and ignore many details of the formalism.

The `hunspell` formalism is used to inflect and derive word forms by a combination of continuation classes, truncation and appending. The formalism resembles the Item and Process morphology (Hockett 1954) and Lexical Phonology (Mohanan 1986). The formalism involves .dic and .aff files that specify the initial word forms and the steps to produce other word forms:

(1) .dic: `glossy/T`
.aff: `SFX T y iest Cy`

The word form *glossiest* is the combination of an input word *glossy*, having the continuation class `T`, and a suffix rule (marked with `SFX`). The suffix rule, for the continuation class `T` (`T` in the 2nd column), states that the last vowel *-y* (`y` in the 3rd column) is replaced with *-iest* (`iest` in the 4th column) if preceded by a consonant and the vowel *-y* (condition `Cy` in the 5th column).

When the `hunspell` dictionary formalism is interpreted as a rewriting system, we see that the derivation `glossyT` $\Rightarrow$ `glossiest#` is described with a Generative Phonological rule (2):

(2) Superlative Formation:

[y]T → [i][e][s][t][#] / C _

While the shape of such a rule is context-sensitive, it is not difficult to see that this rule can be implemented with a non-deterministic finite-state transducer. Furthermore, the suffix rules seem to be applied out from the stem, at the right boundary of the string. However, such similarity with right-linear grammars is only partial and does not imply that it would be easy to compile the whole dictionary into a finite-state transducer. There are two reasons:

– The rules do not only rewrite the continuation classes but they may also back up and rewrite the phonological content produced earlier, requiring, thus, *two-way* movements.
– The rules are *non-monotonic*: they can expand and shorten the input.

In order to analyse what actually happens, we need to construct a model that shows how the dictionary form is processed by the affix rules.

2.2. Automaton models

Now we analyse `hunspell` by viewing the derivation steps of its word formation as a process that corresponds to a computation by a particular TM.

The general definition of a Turing machine is assumed to be familiar to the reader. In short, it is a combination of a finite-state automaton and a rewritable two-way working tape that is initialized with the input string of length n. The machine is allowed to append new letters arbitrarily to the input string; thus the working tape is infinite. A TM can also have auxiliary tapes, but we will restrict ourselves to one-tape TMs.

If we implement the derivation by the moves of a *non-deterministic* one-tape TM, we obtain a machine that sweeps the working tape three times in a row. During the first pass (3.1), the machine recognizes the stem (`glossy`) and its continuation class (`T`), then rewinds the tape (3.2) to the beginning of the string and non-deterministically replaces the substring `syT$$$` with the substring `siest#` during the final pass (3.3):

(3)

$	#	*g*	*l*	*o*	*s*	*s*	*y*	*T*	$	$	$	$	···
→	→	→	→	→	→	→	→	→					(3.1)
	←	←	←	←	←	←	←	←	←				(3.2)
→	→	→	→	→	→	→	→	→	→	→	→		(3.3)
$	#	*g*	*l*	*o*	*s*	*s*	*i*	*e*	*s*	*t*	#	$	···

Since regular relations are closed under composition, a finite number of similar suffix rules could be applied in a row and the lexicon would still be regular. In this way, k suffix positions of morphology could be treated. The corresponding non-deterministic TM would rewind the tape k times to the beginning. Thus, the total time complexity is in $O(nk)$ when the string on the tape occupies at most n tape squares. Thus, the non-deterministic TM implementation of a finite composition has linear time complexity.

In a more general situation, one does not want to specify the maximum number of suffixes explicitly. One reason can be that, in some languages, the suffixes can be added recursively after one other. For example, a Turkish word can, in principle, have an arbitrary number of suffixes although only some of the combinations are interpretable. Another example involves Old Georgian where the nouns can theoretically have unlimited number of case-number markers (Michaelis & Kracht 1997). Finally, orthographic compounding of many languages can involve several stems and alternating bound morphemes. For example, the Swedish word (with our morpheme boundaries) *Spår-vagn-s-aktie-bolag-s-sken-smut-s-skjut-are-fack-förening-s-personal-beklädnad-s-magasin-s-förråd-s-förvaltar-en-s* contains 14 stems and, in addition, several bound morphemes. Similar and much longer examples can be found in other languages. E.g., a 431-letter word appears in the Sanskrit literature. Thus, it is hard to argue that the number of rule applications has a finite upper bound in general.

The unbounded number of applications of suffix rules breaks the two principles: *non-self-embedding rules* and *finite composition*. Moreover, the implementation based on a rewinding TM would spend $O(n^2)$ time to produce the derived string through the back and forth sweeps that simulate the composition steps.

We can, however, improve the TM implementation by optimizing its moves. Instead of rewinding the tape completely, the improved computation strategy (4) just backs up until it has tested the precondition. In our example, the precondition is just the suffix [C][y][T]:

(4)	→	→	→	→	→	→	→	→						(4.1)
	$	#	*g*	*l*	*o*	*s*	*s*	*y*	*T*	$	$	$	$	
							←	←	←					(4.2)
							→	→	→	→	→	→		(4.3)
							s	*i*	*e*	*s*	*t*	#	$	

With this change, long words are produced in a zigzag style (5) where every rule application may back up some letters.

(5) $\rightarrow \quad \rightarrow$

$\hookleftarrow$

$\hookrightarrow \quad \rightarrow$

$\hookleftarrow$

$\hookrightarrow \quad \rightarrow$

$\ldots$

$\hookleftarrow$

$\hookrightarrow \quad \rightarrow$

Since the union of the affix-rules is applied repeatedly to its own output, the standard two-part regularity condition of phonological grammars does not apply. However, as long as the derivation deletes and appends new material only at the right end of the string, the resulting process is linear and, intuitively, a regular grammar. In addition, the moves taken by the TM can now be deterministic because the machine does not completely rewind the tape at any point but always makes relative moves that allow it to remember its previous position.

2.3. Linear encoding

Although the grammar represented by a `hunspell` lexicon does not satisfy the classical two-part condition of finite-state phonology, it is equivalent to a finite-state transducer when restricted to the suffix rules.

There are now some methods to compile `hunspell` lexicons to finite-state transducers. Early experiments on compilation are due to György Gyepesi (p.c., 2007) and others in Budapest. The author developed his solution (Yli-Jyrä 2009) using a variant of Two-Level Morphology (Koskenniemi 1983). This method viewed the lexicon as a collection of constraints that described linearly encoded backing up and suffixation in derivations. The method included an efficient one-shot compilation algorithm to compile and intersect several hundreds of thousands of lexical context restriction rules in parallel as if the lexical continuations (morphotaxis) were phonological constraints. A similar method, finally implemented by his colleagues, Pirinen and Linden (2010), separated the lexical continuations from the phonological changes at morpheme boundaries and used a three-step approach where the final step composed the lexicon with the phonology. A separate compiler for lexical continuations was used and a two-level grammar described the phonological realization of morphemes in different contexts.

The key to understanding the method of Yli-Jyrä (2009) is that the computations of the TM are encoded as a linear string (6.2). This string is produced if the derivation actually reaches the final continuation class # whereas infinite loops do not correspond to an output string.

```
(6)          gloss y                (6.1)
    <#><Root>glossδy<T>δ⁻¹iest<#>   (6.2)
             gloss      iest.       (6.3)
```

Whenever the machine overwrites its own output, the overwritten part (already on the left from the current position) is marked as deleted. The deleted material is put between <D> and <-D>, two symbols abbreviated now as δ and δ^{-1}, respectively. The occurrence of these optional lexical symbols is enforced in deleting contexts but banned otherwise, corresponding to the surface realizations with and without contractions. For example, the segment y in (6.2) is surrounded by a pair of δ and δ^{-1} because it is cancelled by -iest, the next hunspell affix.

The derivation of the combination of glossy and -iest is encoded as string (6.2). This internal string is then mapped to the output string (6.3) by a transducer that deletes the markers and the material enclosed between each pair of δ and δ^{-1}. The computation can also be mapped to the dictionary form (6.1) by removing the markers and the material that belongs to the affixes.

An interesting part in the method is that the underlying derivation encodes a computation of a bounded Turing machine (6.2). This string is produced with context-restriction rules introduced in two-level phonology (Koskenniemi 1983). Since the output deletions are taken care of by the simple transducer between (6.2) and (6.3), it is sufficient to describe only one representation level. The three rules in (7) describe where the root (Root) of the lexicon is visited, where a continuation class T is reached, and how the next (in fact final) continuation class is reached after a consonant, a cancelled y and new material corresponding to lexical -iest.

```
(7) <Root> => <#> _ ;                              # the root symbol
    <T> => <#>δ*<Root>δ*gδ*lδ*oδ*sδ*sδ*yδ* _ ;     # glossy/T
    <#> => Cδy<T>δ⁻¹δ*iδ*eδ*sδ*tδ*<#>δ* _ ;        # SFX T y iest Cy
```

The one-level representation of the underlying derivation works immediately in cases where the successive *deletions* are disjoint from each other. It can also be extended to cases where the deleted parts are nested:

(8)
abcef<A> gh<B> hij<C>
`abcδeδf<A>` δ^{-1}`gh<B>` δ^{-1}`hij<C>`
truncate `f` before `gh<B>`
truncate `egh` before `hij<C>`

The main functional difference between the methods described by Yli-Jyrä (2009) and Pirinen & Lindén (2010) is in the way they treat non-disjoint deletions. While the former method encodes the sequence of derivation steps as one string, the latter encodes the lexical morpheme sequences on one string and then the contracted sequences on the other level. The latter method describes the contractions at morpheme boundaries via two-level rules that constrain the way in which the underlying phonemes of a morpheme are realized in the adjacency of various affixes. In this approach, a contraction corresponds to zero realization.

(9)
abcef<A> gh<B> hij<C>
`abc e f<A> gh<B> hij<C>`
`abc 0 0 0  00 0  hij 0`
truncate `f` before `gh<B>`
truncate `efgh` before `hij<C>`

Since the truncations in this representation (9) are specified in parallel rather than one after another, the semantics of the variant (Pirinen & Lindén 2010) deviates slightly from the original method (Yli-Jyrä 2009).

In particular, note that the addition of the suffix `hij<C>` in (8) and (9) requires different suffixation rules as the truncations behave differently. The rule applied in (9) must truncate more symbols. This semantic difference between the two methods can be compensated with an additional pre-processing step that expands the set of suffixation rules. During this step, a suffix rule that completely cancels the previous affix is replaced with a suffix rule that is applied before the completely cancelled affix. However, for most `hunspell` lexicons, the cancellation is restricted to the most recent suffix, which means that the preprocessing step can be heuristically ignored.

3. The loosest sufficient condition

In the previous section, we related the `hunspell` derivations to one-tape TMs. One reason to do so was that the regularity of one-tape TMs is an old and carefully studied, well-understood problem.

In this section, we first relate one-tape Turing machines with transducers (§3.1–3.2). Then we study bounded one-tape TMs that implement regular relations (§3.3). We will use the bounded one-tape TMs to give a new proof for the two-part regularity condition (§3.4) and to find a more general condition for a finite-state subset of Generative Phonology (§3.5). Finally, we observe (§3.6) that even this condition does not cover all natural finite-state grammars.

3.1. One-tape TMs as transducers

Usually one-tape TMs and Hennie machines are viewed as language recognizers. Since it is not possible to construct a Hennie machine with two readable tapes (Hennie 1965), the connection between Hennie machines and one-way two-tape finite-state transducers is not obvious from the beginning. In fact, most of the relevant literature discusses Hennie machines as if they were equivalent to one-tape finite-state automata only.

As a notable exception, Engelfriet and Hoogeboom (2001) connect Hennie machines to two-way two-tape finite-state machines. These machines are not allowed to read their output tape, but they are more powerful than ordinary finite-state transducers.

Our way to view one-tape Turing machines as transducers requires only the input tape with both reading and writing. As the machine modifies the contents of the input tape during its computations, the input tape will be occupied with an output string when the machine halts. Thus, every one-tape TM recognizes three sets:

– *the set of input strings* that occur as the initial content of the working tape in an accepting computation,

– *the set of output strings* that occur as the final content of the working tape in an accepting computation,

– *the relation consisting of the input-output string pairs* where the first string is the initial content and the second string is the final content of the working tape in an accepting computation.

Given the last definition, every one-tape TM can be viewed as a recognizer of a binary relation.

3.2. Finite-state transducers as one-tape TMs

It is immediate that one-way finite-state transducers are equivalent to one-tape Turing machines: one-way (non)deterministic finite-state transducers are a special case of two-way (non)deterministic finite-state transducers, and these are a special case of (non)deterministic two-tape Turing machines that are equivalent to deterministic one-tape Turing machines.

If we restrict ourselves to finite-state transducers whose output preserves the length of their input, we can view these transducers as one-tape finite automata with a letter-pair alphabet (Kaplan & Kay 1994). This gives an even more direct link from finite-state transducers to one-tape Turing machines.

The restriction to these *letter transducers* is not a serious restriction if we assume that the necessary 0's are introduced non-deterministically to the input string by an inverse homomorphic mapping $h^{-1} : \Sigma^* \to (\Sigma \cup \{0\})^*$ that preserves the alphabet Σ. The 0's are then removed from the output string by the homomorphism $h : (\Sigma \cup \{0\})^* \to \Sigma^*$. In addition, we assume that all states of the transducer have a self-loop on the letter pair $(0, 0)$.

Let R_1 and R_2 be regular relations recognized by unrestricted finite-state transducers, and let R'_1 and R'_2 be the corresponding same-length relations recognized by the letter transducers with the self-loops. Now we have the equation:

$$R_1 \circ R_2 = h^{-1} \circ R'_1 \circ R'_2 \circ h.$$

It is now obvious that the same length relations R'_1, R'_2 and even $R'_1 \circ R'_2$ can be implemented as non-deterministic one-tape TMs that recognize the relations in $O(n)$ time by transforming the initial content of the tape to the final content of the same tape.

3.3. TMs running in $O(n)$ time

The most ground-breaking regularity condition for one-tape TMs is due to Hennie. Hennie's result is the converse to the fact that every one-way deterministic finite automaton is a deterministic TM. The machines considered by Hennie do not only include all one-way deterministic finite automata and letter transducers but they also extend them in two particular ways:

(1) the one-tape TMs can move back and forth on the tape, (2) they can overwrite the contents of the squares of the tape several times.

Hennie (1965) showed that a deterministic one-tape TM is equivalent to a finite automaton if it runs in $O(n)$. The results of Hennie have been extended, by Tadaki et al. (2010), to linear-time non-deterministic one-tape TMs whose $O(n)$ time bound holds for all accepting computations. A deterministic and non-deterministic linear-time one-tape TM are called a *Hennie machine* and a *non-deterministic Hennie machine*, respectively. Both of these one-tape machines recognize a regular relation on the basis of section 3.1.

Hennie analysed the expressive power of one-tape machines using the concept of *crossing sequence* (aka *schema*) (Rabin 1963; Trakhtenbrot 1964; Hopcroft & Ullman 1979; Birget 1996) that is strongly related to *visiting sequences* (Fischer 1969). This concept is a powerful tool in the analysis of the behaviour of two-way automata and one-tape TMs. It refers to the sequence of target states $s_1, s_2, \ldots$ visited by a TM when its pointer crosses the boundary between a pair of adjacent tape squares. States $s_1, s_3, \ldots$ are reached when the pointer moves forward and states $s_2, s_4, \ldots$ are reached when the pointer moves backwards. Figure 1 shows how states are visited during a computation and how a crossing sequence is defined.

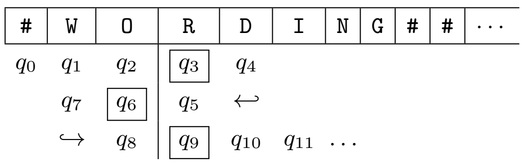


Figure 1: The crossing sequence between the 3rd and the 4th squares is $(s_1, s_2, s_3) = (q_3, q_6, q_9)$

Every Hennie machine satisfies, by definition, the property that the length of its crossing sequences is bounded by an integer (Hennie 1965). Since the finiteness of crossing sequences implies that the TM is equivalent to a finite-state automaton, this bound lets us construct an equivalent finite-state device. The good news is that for all Hennie machines, a bounding constant k is computable (Kobayashi 1985; Tadaki et al. 2010).

Průša (2014) showed that, if a deterministic Hennie machine recognizing the input language has m states and n working symbols, we can construct a minimal deterministic finite automaton that has $2^{2^{O(n \log m)}}$ states. Thus, Hennie machines recognizing the input are much more succinct than the equivalent minimal deterministic automaton. Obviously, a deterministic letter transducer constructed from a Hennie machine is not smaller than the minimal automaton recognizing only the input language aka the domain of the transducer.

3.4. Completeness with respect to prior art

Now we can prove that Hennie machines can be used, on the one hand, to build a finite-state subset of Generative Phonology using the two-part condition of non-self-application and finite composition, and, on the other, to obtain compilation methods for such grammars that previously required specialized encoding and a compilation algorithm.

Theorem 1. *Composition of regular relations is a regular relation.*

Proof. Let R_1 and R_2 be regular relations and R'_1 and R'_2 the respective same-length regular relations. Then there are, respectively, two nondeterministic Hennie machines T_1 and T_2 that recognize R'_1 and R'_2. The composition $R'_1 \circ R'_2$ is then computed by a Hennie machine that first runs like T'_1, then rewinds the tape and runs like T'_2. Since the combined machine preserves the string length, it is equivalent to an epsilon-free finite-state transducer that recognizes the relation $R'_1 \circ R'_2$. The composition $h^{-1} \circ R'_1 \circ R'_2 \circ h$ is then equivalent to the regular relation $R_1 \circ R_2$. □

Theorem 2. *The non-self-embedding application of rule of the form* $\alpha \to \beta/\lambda\_\rho$ *corresponds to a regular relation.*

Proof. Extend the original tape alphabet so that each square contains the input letter and a Boolean vector indicating the validity of left and right context conditions of the simultaneous rules. Let M_L (M_R) be a deterministic (co-deterministic) pattern matching automaton. The state computed by this automaton indicates, for each string position, the type of the prefix (suffix) of the position. Modify this pattern matching automaton by adding self-loops on 0's. Then transform the automaton into a finite-state transducer M'_L (M'_R) in such a way that each transition adds the information on the occurring left (right) contexts to the Boolean vectors of each square. This epsilon-free transducer is a Hennie machine. The composition $M'_L \circ M'_R$ is then a Hennie machine that marks the occurring contexts at all squares.

As a pre-processing step, make the length of each left-hand side α and the respective right-hand side β identical by padding the shorter with 0's.

In addition, add synchronous 0's freely to both. In this way we obtain a letter transducer that recognizes a 0-padded representation of the regular relation $\alpha \times \beta$.

Define a Hennie machine M_1 that sweeps the string (containing 0's) from left to right and non-deterministically overwrites ranges of squares that contain some left-hand-side string α with a corresponding right-hand-side string β when the first and the last square in the input range indicate the presence of the required left and right context, respectively. Define also a Hennie machine M_2 that removes the Boolean context vectors from the tape squares. Now the composition $M'_L \circ M'_R \circ M_1 \circ M_2$ is recognized by a Hennie machine M. Then $h^{-1} \circ M \circ h$ is equivalent to a non-deterministic finite-state transducer that captures the semantics of the rule. □

We have now used Hennie machines to show that simultaneous *non-overlapping* rules are regular and that a *finite composition* of regular rules preserves regularity. Other application modes of regular grammars are discussed in Johnson (1972); Kaplan & Kay (1994). The regularity of these application modes can be proven similarly.

To conclude our argument, we show that Hennie machines actually help us to compile `hunspell` dictionaries without special encodings.

Theorem 3. *The iterated application of monotonic suffix rules of a* `hunspell` *grammar makes a regular relation.*

Proof. Every suffix rule corresponds to a Hennie machine that backs up checking its context condition and then writes the non-truncated context and the new suffix (4.2–4.3). The union of such Hennie machines is a non-deterministic Hennie machine M. The closure M^* is a TM that applies suffix rules iteratively. As the suffix rules increase the length of the string monotonically, the closure M^* has a finite bound for the crossing sequences and recognizes a regular relation. □

3.5. The bound that cannot be improved

The Borodin-Trakhtenbrot Gap Theorem (Trakhtenbrot 1964) states that expanded resources do not always expand the set of computable functions. In other words, it is possible that the regularity of a TM holds even if the $O(n)$ is made slightly looser. A less tight time bound is now expressed with the small-o notation: $t(n) \in o(f(n))$ means that the upper bound $f(n)$ grows much faster than the running time $t(n)$ when n tends to infinity: $\lim_{n\to\infty} t(n)/f(n) = 0$.

As an application of the Gap Theorem, Hartmanis (1968) and Trakhtenbrot (1964) showed independently that the time resource of finite-state equivalent deterministic one-tape TMs can be extended from $O(n)$ to $o(n \log n)$. This bound is tight: regularity is algorithmically unsolvable for any bound that exceeds $n+1$ in $\Omega(n \log n)$ (Gajser 2015). The extended time bound has been generalized to non-deterministic one-tape TMs by Tadaki et al. (2010). These extensions of Hennie's core result give us a new sufficient condition for the regularity of Generative Phonology.

Theorem 4. *A generative phonological grammar is regular if its one-tape TM implementation runs in* $o(n \log n)$ *time.*

Let M be a one-tape TM implementation of a Generative phonological grammar. The finiteness of the crossing sequences of a given TM is, in general, undecidable (Průša 2014), but there is a reasonably good decision procedure: to test if M is equivalent to a finite-state transducer, we can pick a function $t(n)$ that is in $o(n \log n)$ and test if M actually runs in $t(n)$. Interestingly, Gajser (2015) showed that for any reasonable function $t(n)$, we can decide whether a TM M runs in $t(n)$. If a TM then runs in $t(n)$, it actually runs in $O(n)$ (Pighizzini 2009). Thus, the new one-sided condition for regularity of the phonological grammar has a sound approximate solution.

3.6. The existence of non-Hennie finite-state grammars

If the suffix rules are non-monotonic and can shorten their inputs, the TM can produce the same configuration again and again and produce arbitrarily long crossing sequences. The repetition may happen either a finite or an infinite number of times. Interestingly, the specialized compilation method (Yli-Jyrä 2009) handles both cases correctly, whereas we fail to get a Hennie machine if the suffix rules are non-monotonic.

Non-monotonic suffix rules are an example of a situation where the TM is equivalent to a Hennie machine that restricts the length of the crossing sequences. The bad news is that we do not know when we have a correct Hennie machine: it is difficult to find such bound k for the length of crossing sequences that a given TM preserves its semantics when longer crossing sequences are abandoned. Since a sufficient bound k is such that the semantics of the restricted TM does not change although we allow longer crossing-sequences, there are reasonable ways to probe possible values of k, but such probing is still heuristic.

The difficulty of non-monotonic grammars indicates that although we now have a more general condition for those Generative phonological grammars that are equivalent to a finite-state transducer, a specialized compilation algorithm may still encode infinite loops in a way that seems to be beyond the Hennie condition.

4. Conclusions

It is historically interesting that Hennie's regularity condition dates back to the year 1965, that is, even before Chomsky & Halle (1968).

No decision procedure for the classical two-part condition (Johnson 1972; Kaplan & Kay 1994), is known. Compared to this situation, it is remarkable that the new sufficient condition has several advantages:

- The new regularity condition has approximations that are decidable (Gajser 2015).
- The equivalent finite-state transducer can be constructed from a Hennie machine (Hennie 1965).
- The Hennie machines are extremely succinct compared to finite-state machines (Průša 2014).
- Hennie machines seem to provide a more general framework for proving regularity of phonological grammars than the arguments based on bimachine construction (Johnson 1972) or non-self-embedding grammars (Kaplan & Kay 1994).

There are many interesting questions that could be studied in the future. Here are some:

1. Despite the advantages of Hennie machines, the author is not aware of any finite-state library that would be based on Hennie machines. *Would it be possible to develop a finite-state library that would use Hennie machines to represent regular relations more compactly?*
2. There does not seem to be much work that would link Hennie machines and two-way finite-state transducers to *minicomplexity*, the computational complexity of two-way finite automata, that has recently obtained attention in automata theory (Kapoutsis 2012). *Could some of the related results be extended to Hennie machines?*
3. If a Hennie machine is used to implement a weighted rule system, the machine must be constructed more carefully than what we have done now: the 0-loops create new paths that make the computation

of string weights tricky. *Can we introduce weighted Hennie machines and relate them to weighted automata?*

4. We would like to understand why some natural finite-state grammars, like nonmonotonic `hunspell` grammars, are finite-state, although their crossing sequences seem to have no finite bound. *Are there thus other natural islands of regularity we should know about?*

There are several potential applications for Hennie machines in Natural Language Processing. We have already demonstrated that Hennie machines have applications in phonology and morphology (Yli-Jyrä 2009). Weighted Hennie machines may be applied to OCR that is based on weighted context-dependent correction rules (Drobac et al. 2017). Furthermore, non-monotonic Sequential Constraint Grammar is computationally undecidable but has restrictions that have Hennie machine characterizations (Yli-Jyrä 2017). The search for Generative dependency grammars that produce non-projective trees is an area that may also benefit from the concepts of crossing sequences and Hennie machines (Nederhof & Yli-Jyrä 2017).

Acknowledgements

Writing this article was supported by the author's fellowship #270354 from the Academy of Finland.

References

Beesley, K. R. and L. Karttunen. 2003. Finite state morphology. Stanford, CA: CSLI Publications.

Birget, J.-C. 1996. Two-way automata and length-preserving homomorphisms. Mathematical Systems Theory 29. 191–226.

Chomsky, N. 1963. Formal properties of grammars. In R. Luce, R. Bush and E. Galanter (eds.) Handbook of mathematical psychology. New York: John Wiley and Sons. 323–418.

Chomsky, N. and M. Halle. 1968. The sound pattern of English. New York: Harper & Row.

Drobac, S., P. Kauppinen and K. Lindén. 2017. OCR and post-correction of historical Finnish texts. In Proceedings of the 21st Nordic Conference of Computational Linguistics. Gothenburg: Linkoping University Electronic Press, 70–76. http://www.ep.liu.se/ecp/131/009/ecp17131009.pdf

Engelfriet, J. and H. J. Hoogeboom. 2001. MSO definable string transductions and two-way finite-state transducers. ACM Transactions on Computational Logic 2. 216–254.

Fischer, M. J. 1969. Two characterizations of the context-sensitive languages. In IEEE Conference Record of 10th Annual Symposium on Switching and Automata Theory. 149–156. http://ieeexplore.ieee.org/document/4569611/

Gajser, D. 2015. Verifying time complexity of Turing machines. Theoretical Computer Science 600. 86–97.

Greibach, S. 1968. A note on undecidable properties of formal languages. Mathematical Systems Theory 2. 1–6.

Hartmanis, J. 1968. Computational complexity of one-tape Turing machine computations. Journal of the ACM 15. 325–339.

Hennie, F. C. 1965. One-tape, off-line Turing machine computations. Information and Control 8. 553–578.

Hockett, C. F. 1954. Two models of grammatical description. Word 10. 210–231.

Hopcroft, J. E. and J. D. Ullman. 1979. Introduction to automata theory, languages and computation. Reading, MA: Addison-Wesley.

Johnson, C. D. 1972. Formal aspects of phonological description. The Hague: Mouton.

Jurafsky, D. and J. H. Martin (eds.). 2000. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech processing. Englewood Cliffs, NJ: Prentice-Hall.

Kaplan, R. M. and M. Kay. 1994. Regular models of phonological rule systems. Computational Linguistics 20. 331–378.

Kapoutsis, C. A. 2012. Minicomplexity. In M. Kutrib, N. Moreira and R. Reis (eds.) Descriptional Complexity of Formal Systems: 14th International Workshop, DCFS 2012, Braga, Portugal, July 23–25, 2012. Proceedings. Berlin & New York: Springer. 20–42.

Karttunen, L. 1995. The replace operator. In 33rd Annual Meeting of the Association for Computational Linguistics. 16–23. https://dl.acm.org/citation.cfm?id=981658.981661

Kobayashi, K. 1985. On the structure of one-tape nondeterministic Turing machine time hierarchy. Theoretical Computer Science 40. 175–193.

Koskenniemi, K. 1983. Two-level morphology: A general computational model for word-form recognition and production. Helsinki: University of Helsinki.

McCarthy, J. J. 2000. Harmonic serialism and parallelism. In M. Hirotani, A. Coetzee, N. Hall and J.-Y. Kim (eds.) Proceedings of NELS 30. Amherst: University of Massachusetts. 501–524.

Michaelis, J. and M. Kracht. 1997. Semilinearity as a syntactic invariant? Logical Aspects of Computational Linguistics 1328. 329–345. https://doi.org/10.1007/BFb0052165

Mohanan, K. P. 1986. The theory of lexical phonology. Dordrecht: D. Reidel.

Mohri, M. and R. Sproat. 1996. An efficient compiler for weighted rewrite rules. In 34th Annual Meeting of the Association for Computational Linguistics. 231–238. http://aclweb.org/anthology/P96-1031

Nederhof, M.-J. and A. Yli-Jyrä. 2017. A derivational model of discontinuous parsing. In Language and Automata Theory and Applications – 11th International Conference, LATA 2017, Umeå, Sweden, March 6–9, 2017, Proceedings. 299–310. https://doi.org/10.1007/978-3-319-53733-7_22

Németh, L., V. Trón, P. Halácsy, A. Kornai, A. Rung and I. Szakadát. 2004. Leveraging the open source ispell codebase for minority language analysis. In J. Carson-Berndsen (ed.) Proceedings of the SALTMIL Workshop at LREC 2004. 56–59.

Pighizzini, G. 2009. Nondeterministic one-tape off-line Turing machines and their time complexity. Journal of Automata, Languages and Combinatorics 14. 107–124.

Pirinen, T. A. and K. Lindén. 2010. Creating and weighting Hunspell dictionaries as finite-state automata. Investigationes Linguisticae 21. 1–16.

Popper, K. R. 1959. The logic of scientific discovery (Second edition). London: Hutchinson & Co.

Prince, A. and P. Smolensky. 2004. Optimality Theory: Constraint interaction in Generative Grammar. Malden, MA & Oxford: Blackwell.

Průša, D. 2014. Weight-reducing Hennie machines and their descriptional complexity. In A.-H. Dediu, C. Martín-Vide, J.-L. Sierra-Rodríguez and B. Truthe (eds.) Language and Automata Theory and Applications: 8th International Conference, LATA 2014, Madrid, Spain, March 10–14, 2014. Proceedings. Cham: Springer. 553–564.

Rabin, M. O. 1963. Real time computation. Israel Journal of Mathematics 1. 203–211.

Ristad, E. S. 1990. Computational structure of generative phonology and its relation to language comprehension. In Proceedings of the 28th Annual Meeting on Association for Computational Linguistics. 235–242. http://dx.doi.org/10.3115/981823.981853

Stearns, R. E. 1967. A regularity test for pushdown machines. Information and Control 11. 323–340.

Tadaki, K., T. Yamakami and J. C. H. Lin. 2010. Theory of one-tape linear-time Turing machines. Theoretical Computer Science 411. 22–43.

Trakhtenbrot, B. A. 1964. Turing computations with logarithmic delay (in Russian). Algebra i Logika 3. 33–48. English translation in U. of California Computing Center, Tech. Report. No. 5, Berkeley, CA, 1966.

Yli-Jyrä, A. 2009. An efficient double complementation algorithm for superposition-based finite-state morphology. In K. Jokinen and E. Bick (eds.) NODALIDA 2009 Conference Proceedings. 206–213. http://dspace.ut.ee/handle/10062/9765

Yli-Jyrä, A. 2017. The power of constraint grammars revisited. In Proceedings of the NoDaLiDa 2017 Workshop on Constraint Grammar – Methods, Tools and Applications, 22 May 2017, Gothenburg, Sweden. 23–31. http://www.ep.liu.se/ecp/140/005/ecp17140005.pdf

K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 561–582

On raised verb phrases

RICHARD ZUBER
CNRS, Laboratoire de Linguistique Formelle, Paris
Richard.Zuber@linguist.univ-paris-diderot.fr

KEYWORDS

type raising
generalized NPs
raised VPs
higher order invariance principles
non-homomorphic predicates

ABSTRACT

Raised verb phrases denote elements of $\wp(\wp(\wp(E)))$, that is sets of type $\langle 1 \rangle$ quantifiers (and not just sets). Various arguments supporting the necessity of the VP raising, similar to the noun phrase raising, are given. Most of the presented arguments are related to the semantics of the higher order comparative *the same* and the semantics of the reciprocal *each other* but some other constructions with raised VPs are also discussed. Predicates formed by such constructions are "non-homomorphic" because they denote sets of quantifiers whose characteristic functions are not homomorphisms (from the algebra of quantifiers to the algebra of truth-values). Some formal properties and analogies with "classically" raised NPs are indicated.

1. Introduction

The results of combinatorial logic allow us to abandon, in certain cases, the distinction between an argument and the function of which it is the argument: informally, the argument of a function can become the function having as argument the function of which it was the argument. More formally, in the categorial grammar that includes "functional categories" and where grammatical categories are associated with logical types, an expression can be associated with at least two types: if it occurs "initially" in type a it may also occur in any type $\langle\langle a, b\rangle, b\rangle$ for any type b. Probably having this in mind, Montague made his by now well-known move which led to the uniform treatment of noun phrases which all, including proper nouns, denote sets of properties. In a categorial grammar in which NP and S are primitive categories, and ignoring directionality, Montague's idea can be illustrated at the syntactic level, by the fact that the sequences of categories in (1) and (2) reduce, *via* the function application (symbolised by "$+$"), to the same category S:

(1) $NP + S/NP{=}S$

(2) $S/(S/NP) + S/NP{=}S$

At the semantic level, adding (2) to the grammar amounts to considering that denotations of proper nouns, which "classically" denote individual objects, that is objects of type e, get a "new" denotation which is the ultrafilter generated by the element of the model corresponding to the referent of the proper noun and is now of type $\langle\langle e,t\rangle,t\rangle$. This move makes it easy in particular to compute the semantic values of Boolean compounds of proper nouns with other NPs since in this case Boolean connectors are interpreted by the corresponding Boolean operations.

In (1) the first element is considered as an argument expression and the second as the functional expression. In (2) the roles are inverted: the first element is the functional expression and the second is the argument expression. Thus in (2) a type-raising rule, generally admitted in categorial grammars, has been applied: this rule turns arguments into functions over functions over these arguments.

Type-raising is one of the tools used in the strategy of flexible categories. It amounts to the proposal that some linguistic units identified by the (categorial) grammar may have many categories associated with them and thus take their denotations among various logical types. As seen from (1) and (2), type-raising is related to the rule of function application. Other syntactic rules, such as for instance function composition (cf. Geach 1972), can be used to define other type changing operators that enrich the tools allowing type shifting (cf. Partee & Rooth 1983; Partee 1987, for some empirically justified pioneering proposals).

Obviously, from a theoretical point of view there is no reason for this process of inversion of roles between a function and its argument to stop: the reduction indicated in (3) is also possible:

(3) $S/(S/NP){+}\ S(S/(S/NP)) = S$

In (2) the category of the subject NP has been raised to the functional category $S(S/NP))$ and in (3) the verb phrase, which was the argument expression of category S/NP in (2) becomes functional expression of category $S(S/(S/NP))$. Thus, in (3) the VP has been raised to the category $S(S/(S/NP))$ whose type is now $\langle\langle\langle e,t\rangle t\rangle t\rangle$. This means that such raised VPs denote a set of type $\langle 1\rangle$ quantifiers and consequently the sentence of the form (4), where VP^R is a raised VP (that is, VP^R is the abbreviation of the category $S(S/(S/NP)))$, is true iff the quantifier denoted by the NP belongs to the set denoted by the VP^R:

(4) $NP + VP^R$

Of course VP^R is a verb phrase. It gets an additional category, that is, VP^R is an abbreviation of $S(S/(S/NP))$ and is interpreted now by objects of type $\langle\langle\langle e,t\rangle t\rangle t\rangle$, that is, it denotes elements of $\wp(\wp(\wp(E)))$. Consequently, given this alternative, verb phrases act as functions taking subject NPs (which in this case necessarily denote type $\langle\langle e,t\rangle,t\rangle$ objects) as arguments.

In this article I give some empirical reasons in favour of adding a rule like the one in (3) to the grammar. In other words, I will indicate a series of linguistic data which can be uniformly treated in the framework in which it is assumed that in addition to the subject NPs also verb phrases have to be raised. To do this I will discuss the semantics of some specific linguistic constructions which *induce* or *force* the raising of VPs to which they are related. Consequently, at the semantic level, I will show that it is useful, if not necessary, to suppose that in some cases verb phrases denote sets of type $\langle 1\rangle$ quantifiers.

The idea of type shifting is that the type of some categories can change depending on the environment they find themselves in. This means that the type of a given category has to be changed only in some grammatical constructions. A consequence of this is the fact that lifted VPs are not morphologically or syntactically "simple" since they are usually results of various operations due to lift inducers, sometimes language specific. Probably for this reason, the proposal that languages might differ from each other as to whether it is the subject NP or the VP that takes takes the other as argument is not new. In addition, some specific linguistic phenomena may be better treated in such an extended framework. For instance, Bach (1980) relates the difference between tensed and untensed intransitive VPs precisely to the difference in types associated with them. Similarly, it has been occasionally suggested that the specific "plural verbs" such as *to gather* or collective predicates such as *meet*, have their type raised and thus that they denote elements of $\wp(\wp(E))$ (van der Does 1993). However, as far as I know, this VP raising always goes in pair with type shifting of nominal elements, in particular of determiners, as well. For van der Does (1993) the ordinary determiners which "classically" are relations between sets, get an additional type making them relations between sets and sets of sets.

A special case of VPs with higher type is discussed in Partee & Rooth (1983). This case is special since it involves intensionality explicitly: in order to account for the semantics of complex VPs, which are conjunctions of intensional and extensional verbs, they have to be raised and get a higher order type.

More generally, Keenan and Faltz (1985) propose that (extensional) VPs always denote specific characteristic functions of a set of type $\langle 1 \rangle$ quantifiers, that is, they denote a set of quantifiers. Given that these characteristic functions are homomorphisms in addition (from the algebra of quantifiers to the algebra of truth-values), they indicate that the denotational domain of VPs that they propose is isomorphic to the algebra of sets (subsets of the universe), the classical denotational domain of one-place predicates in first order logic. At the same time, Keenan and Faltz (1985, 265) indicate that in natural languages there exist various "non-homomorphic" predicates such as collective and reciprocal predicates, which cannot denote in the denotational domain of VPs that they propose.

In this paper I argue for changing the denotational type of VPs. I will mainly discuss complex verb phrases containing transitive verbs whose second argument, the argument in the object position, is what I will call a generalised noun phrase (GNP), that is an expression which can play the role of nominal arguments of a verb, as does an ordinary NP, but which cannot freely occur in all argumental positions of the verb (cf. Zuber 2018). These expressions will be characterised by their logical properties and not by their syntactic properties. A typical example of such a GNP is the reciprocal *each other* and various Boolean compounds of it with ordinary NPs or reflexive NPs. Another example that will be discussed at some length is the "higher order" comparative like *the same CN* or *the same number of CN*. It will be indicated that such GNPs, which force the raising of VPs, have various logical properties that differentiate them from ordinary NPs in the object position.

The second series of constructions I will discuss concerns raised VPs formed from intransitive VPs. Such VPs, whose semantics necessitates raising, can be either simple intransitive VPs or complex VPs with the intransitive verb modified by specific adverbials or gerundives which induce the raising. As we will see, such adverbials are usually semantically related to GNPs. In this context I will mention a possible analysis of cumulative readings of some quantifiers and some other readings related to the plurality of subject NPs, in which the rule of VPs raising is explicitly used. In fact it will appear that cumulative readings (of NPs in subject and object positions) can be, or even should be, related to the semantics of the GNPs such as *the same* or *each other*. More generally, it will appear that many expressions forcing the raising of VPs are semantically related, and some of them, roughly speaking, are defined by others, at least at up to some "degree of equivalence".

Finally, I will recall that, as it is the case with proper nouns when they occur in conjunction with quantified NPs, "Booleanly" simple VPs, given by intransitive verbs, which are of type $\langle e, t\rangle$, must have their type raised when they occur in conjunctions with (simple or complex) VPs whose type is raised. Some other similarities with the NP raising will be indicated and in particular the existence of the inverse rule of VP lowering. For this reason we will call VP raising *classical raising*.

To conclude these introductory remarks I want to stress that the purpose of this paper is not to give a full or detailed semantics of the constructions that will be mentioned. I will discuss essentially examples of syntactically complex constructions whose semantics has already been specified precisely in the spirit of the proposal made in this article. Indeed, it seems obvious that VPs may non-trivially denote types other than that of sets only when they form syntactically complex constructions.

2. Formal preliminaries

We will consider binary relations and functions, in particular type $\langle 1\rangle$ quantifiers, over a universe E. To note the type of function we will use not only Montagovian notation. In particular the type of functions from binary relations to sets of type $\langle 1\rangle$ quantifiers will be noted $\langle 2 : \langle 1\rangle\rangle$ and the type of functions having binary relations and sets as arguments and sets of type $\langle 1\rangle$ quantifiers as output will be noted $\langle 2, 1 : \langle 1\rangle\rangle$.

If R is a binary relation, $D(R)$ denotes its domain.The relation I_d is the identity relation: $I_d = \{\langle x, y\rangle : x = y\}$. If R is a binary relation and X a set then $R/X = R \cap (X \times X)$. The binary relation R^S is the greatest symmetric relation included in R, that is $R^S = R \cap R^{-1}$ and $R^{S-} = R^S \cap I'_d$ is the greatest symmetric irreflexive relation included in R. For any binary relation R and any set A, the relation R_A is the subset of R defined as $R_A = \{\langle x, y\rangle : \langle x, y\rangle \in R \wedge y \in A\}$.

Let Q be a type $\langle 1\rangle$ quantifier. Q is atomic iff it is a singleton. An atomic quantifier containing A as its only element will be noted Q_A. Q is positive, $Q \in POS$ iff $\emptyset \notin Q$; Q is natural iff either Q is positive and $E \in Q$ or Q is not positive and $E \notin Q$. Two natural quantifiers have the same polarity iff either both are positive or neither of them is positive.

We will also use the property *living on* displayed by type $\langle 1\rangle$ quantifiers (cf. Barwise & Cooper 1981). The type $\langle 1\rangle$ quantifier Q lives on a set A (where $A \subseteq E$) iff for all $X \subseteq E$, $Q(X) = Q(X \cap A)$. If E is finite then there is always a smallest set on which a quantifier Q lives. The fact that A is a set on which Q lives will be noted $Li(Q, A)$ and the fact that A is a smallest set on which Q lives will be noted $SLi(Q, A)$. If $Li(Q, A)$ and $B \subseteq A \wedge B \in Q$ then B is a *witness set* of Q. The fact that B is a witness set of the quantifier Q, which lives on A, will be noted $B = Wt(Q, A)$. If $Li(Q, A)$ then $A \in Q$ iff $E \in Q$ and thus if $E \in Q$ and $Li(Q, A)$ then $A = Wt(Q, A)$.

Observe that any principal filter is a positive type $\langle 1 \rangle$ quantifier that lives on the set by which it is generated, and, moreover, this set is its witness set. Atomic quantifiers live on the universe E only.

Concerning syntactic aspects we will use a "simple extended categorial grammar" admitting flexible categories. Thus we assume that for each derived category C of the form $C = A/B$ there is a rule stating that an expression of category A can be built by combining an expression of category B with an expression of category C. For any grammatical category C there is a corresponding denotational Boolean algebra D_C of possible denotations of expressions of category C. Expressions of the derived category A/B take their denotations in the algebra $D_{A/B}$ which is the algebra of functions from D_B to D_A. Furthermore, given that most categories are functionally related (in principle all "major" categories are Boolean), the corresponding denotational algebras are not independent of each other. In particular the elements of the algebra $D_{A/B}$ are functions from D_B to D_A. Given that functions interpreting functional expressions in general satisfy various constraints, one usually considers just some sub-algebras of the algebra of all functions from D_B to D_A. For instance NPs denote in the algebra $D_{S/VP}$ of type $\langle 1 \rangle$ quantifiers.

Among type $\langle 1 \rangle$ quantifiers we distinguish *nominal individuals* I_a defined as $I_a = \{Y : Y \subseteq E \wedge a \in Y\}$. Nominal individuals are denotations of proper nouns. They are obtained precisely by the operation of type raising applied to (denotations of) proper nouns "initially" having as denotation objects of type e. Nominal individuals belong to the class of quantifiers called *principal filters generated by a set*. Thus $Ft(A)$, the (principal) filter generated by the set A (for $A \subseteq E$), is defined as: $Ft(A) = \{Y : Y \subseteq E \wedge A \subseteq Y\}$.

The notion of an individual can in fact be associated with any Boolean denotational algebra:

D1: Let B be an atomic Boolean algebra and $I \subseteq B$. Then I is an individual on B iff χ_I, the characteristic function of I, is a homomorphism from B to the algebra $\{0, 1\}$

Nominal individuals are individuals on the algebra of sets in the sense of D1.

An individual I on B is generated by the atom α of B iff $\alpha \leq i$ for any $i \in I$. Individuals of an atomic algebra B, generated by an atom of B, are thus exactly the sets of elements of B which satisfy (1) the meet, (2) the join and (3) the complement conditions. More formally, if I is an individual (on the algebra B, generated by an atom of B) then for any $S \subseteq B$ we have (1) $S \subseteq I$ iff $\bigwedge S \in I$, (2) $S \cap I \neq \emptyset$ iff $\bigvee S \in I$, and (3) $\alpha \in I$ iff $\alpha' \notin I$, for any $\alpha \in B$ (where "$\bigvee$" and "$\bigwedge$" denote arbitrary meets and joins respectively, in B).

The denotation of the expression α will be noted $[\alpha]$ and we will be interested only in the extensional aspects of the meaning. If α is a VP which denotes the set P, a subset of the universe, then α^R, raised α, denotes a set of type $\langle 1 \rangle$ quantifiers:

(5) $[\alpha^R] = \{Q : Q(P) = 1\}$, where Q is a type $\langle 1 \rangle$ quantifier.

The set of type $\langle 1 \rangle$ quantifiers, associated with the property P, defined in (5) is particular because its characteristic function is a homomorphism from the algebra of type $\langle 1 \rangle$ quantifiers to the algebra of truth values. It follows from this that the set in (5) corresponds to the individual on the algebra D_{NP} generated by the atomic quantifier Q_P. Such individuals, that is individuals on the algebra D_{NP} generated by atomic (type $\langle 1 \rangle$) quantifiers will be called *verbal individuals*. One can see that any verbal individual has at least one nominal individual as a member. Furthermore, a verbal individual is in particular a complete set of quantifiers (every type $\langle 1 \rangle$ quantifier or its Boolean complement belong to the set) and it is consistent (no quantifier and its Boolean complement belong to it).

Given the fact that the denotational algebras of (non-raised) VPs and of characteristic functions of verbal individuals are isomorphic we can say that "classically" VPs denote (up to the isomorphism) verbal individuals. In this paper we consider a more general case: we suppose that there is the denotational algebra D_{VP^R}, which is the set of functions from D_{NP} to the algebra $\{0, 1\}$, and these functions need not to be homomorphisms. This lack of homomorphism property will be the basic semantic property of the constructions that will be considered.

Since our basic argument for the necessity of raised verb phrases uses transitive VPs with special direct objects we need to specify how the composition between the transitive verb and its second argument, the direct object, is realised. I will follow here the well-justified proposal in Keenan (2016) who indicates various merits of the interpretation of the direct objects *in situ*, as functions taking binary relations, denotations of transitive verbs, as arguments. Thus, Keenan proposes that direct object NPs are of the category $(S/NP)/((S/NP)/NP)$. Formally, at the semantic level, this is done by extending the domain of type $\langle 1\rangle$ quantifiers: in addition to sets, the basic domain of type $\langle 1\rangle$ quantifiers, relations are also considered as their possible arguments. Thus type $\langle 1\rangle$ quantifiers, considered as functions, can apply not only to sets but also additionally to relations, denotations of transitive (ditransitive, etc.) verbs. When such functions with the extended domain act as denotations of direct objects, they are accusative extensions Q_{acc} of the quantifier Q, defined in D2 (i), and when they act as denotations of subjects (NPs in nominative case) of transitive sentences they are nominative extensions defined in D2 (ii):

D2 (i): For each type $\langle 1\rangle$ quantifier Q, $Q_{acc}R = \{a : Q(aR) = 1\}$
(ii): For each type $\langle 1\rangle$ quantifier Q, $Q_{nom}R = \{a : Q(Ra) = 1,$
where $aR = \{y : \langle a, y\rangle \in R\}$ and $Ra = \{y : \langle y, a\rangle \in R\}$.

The nominal extension of a quantifier can be used to represent readings of transitive sentences with the object taking wide scope (Keenan 2016).

Nominal and accusative case extensions are specific type $\langle 2 : 1\rangle$ functions. One can distinguish various kinds of type $\langle 2 : \langle 1\rangle\rangle$ and type $\langle 1, 2 : \langle 1\rangle\rangle$ functions. Observe first that any type $\langle 2 : 1\rangle$ function whose output is denoted by a (non-raised) VP can be lifted to a type $\langle 2 : \langle 1\rangle\rangle$ function. The accusative extension of a type $\langle 1\rangle$ quantifier Q can be lifted to a type $\langle 2 : \langle 1\rangle\rangle$ function in the way indicated in (6). Such functions will be called *accusative lifts*. More generally, if F is a type $\langle 2 : 1\rangle$ function, its lift F^L, a type $\langle 2 : \langle 1\rangle\rangle$ function, is defined in (7):

(6) $Q^L_{acc}(R) = \{Z : Z(Q_{acc}(R)) = 1\}$.

(7) $F^L(R) = \{Z : Z(F(R)) = 1\}$.

The variable Z above ranges over the set of type $\langle 1\rangle$ quantifiers.

We will also use two types of set partitions, defined by the binary relation R. First, if R is an irreflexive symmetric relation (i.e., $R \cap R^{-1} \cap I_d = \emptyset$) then $\Pi(R)$ is the least fine partition of R such that each of its blocks is of the form $(A \times A) \cap I'_d$. A partition is *trivial* iff it contains only one block. Observe that if R is an irreflexive symmetric relation and $\Pi(R)$ is not trivial, then every block of $\Pi(R)$ contains at least two elements.

Second, to analyse the sentences with *the same CN* and *the same number of CN* we will use partitions induced by the following equivalence relations associated with the binary relation R:

D3 (i): $e_R = \{\langle x, y\rangle : xR = yR\}$
(ii) $e_{R,n} = \{\langle x, y\rangle : |xR| = |yR|\}$

To show that it is necessary to raise the type of VPs to get the right semantics of some constructions I will indicate some semantic properties of these constructions and show that they are incompatible with the properties held by non-raised VPs. For non-raised VPs the following is true: sentences of the form in (8a) are equivalent to sentences of the form (8b):

(8) a. $(NP_1\ VP)$ *and* $(NP_2\ VP)$
b. $(NP_1$ *and* $NP_2)\ VP$

In other words if NP_1 denotes the quantifier Q_1, NP_2 denotes the quantifier Q_2 and VP denotes the property P then (9) holds:

(9) $(P \in Q_1 \wedge P \in Q_2) \equiv P \in (Q_1 \cap Q_2)$

The property in (9) is a consequence of the fact that quantifiers denoted by the subject NPs are homomorphisms from the algebra of sets (subset of a given universe) to the algebra of truth values. This property will be frequently used as a test to check whether a certain type of a VPs denotes a set. It will be called *homomorphism test* or h-test.

In the same way, for type $\langle 2 : \langle 1\rangle\rangle$ functions which are lifts of type $\langle 2 : 1\rangle$ functions we have:

Proposition 1: If a type $\langle 2 : \langle 1\rangle\rangle$ function F is a lift of a type $\langle 2 : 1\rangle$ function then for any type $\langle 1\rangle$ quantifiers Q_1 and Q_2 and any binary relation R, if $Q_1 \in F(R)$ and $Q_2 \in F(R)$ then $(Q_1 \wedge Q_2) \in F(R)$

Accusative lifts satisfy the following higher order extension condition *HEC* (Zuber 2014):

D4: A type $\langle 2 : \langle 1\rangle\rangle$ function F satisfies HEC (higher order extension condition) iff for any natural type $\langle 1\rangle$ quantifiers Q_1 and Q_2 with the same polarity, any $A, B \subseteq E$, any binary relations R, S, if $Li(Q_1, A)$, $Li(Q_2, B)$ and $\forall_{a\in A}\forall_{b\in B}(aR = bS)$ then $Q_1 \in F(R)$ iff $Q_2 \in F(S)$.

For functions satisfying HEC we have:

Proposition 2: Let F satisfies HEC and let $R = E \times C$, for $C \subseteq E$ arbitrary. Then for any $X \subseteq E$ either $Ft(X) \in F(R)$ or for any X, $Ft(X) \notin F(R)$.

Thus, a function satisfying the HEC condition, whose argument is the cross-product relation of the form $E \times A$ has in its output either all principal filters or no principal filter. Thus Proposition 2 can be used to show that the function denoted by *each other* and by other expressions that induce the VP raising do not satisfy HEC. Functions denoted by such expressions satisfy conditions which are strictly weaker than HEC. Thus the denotations of higher order anaphors satisfy the *higher order predicate invariance* or *HPI*. By definition (Zuber 2014):

D5: A type $\langle 2 : \langle 1 \rangle \rangle$ function F satisfies HPI (higher order predicate invariance) iff for a type $\langle 1 \rangle$ quantifier Q, any $A \subseteq E$, any binary relations R, S, if $Li(Q, A)$ and $\forall_{a \in A}(aR = aS)$ then $Q \in F(R)$ iff $Q \in F(S)$.

An equivalent way to define HPI is as follows:

Proposition 3: Function F satisfies HPI iff $Li(Q, A)$ entails $Q \in F(R)$ iff $Q \in F((A \times E) \cap R)$.

Similarly, higher order comparatives satisfy the so-called *higher order argument invariance* or *HAI* (Zuber 2014):

D6: A type $\langle 2 : \langle 1 \rangle \rangle$ function F satisfies HAI (higher order argument invariance) iff for any natural type $\langle 1 \rangle$ quantifiers Q_1 and Q_2 with the same polarity, any $A, B \subseteq E$, any binary relation R, if $SLi(Q_1, A)$, $SLi(Q_2, B)$ and $\forall_{a \in A} \forall_{b \in B}(aR = aS)$ then $Q_1 \in F(R)$ iff $Q_2 \in F(R)$.

Obviously HEC entails both HPI and HAI.

3. Generalized noun phrases and raised verb phrases

The first class of VP raising inducers we discuss, in some sense the most important one, is represented by proper GNPs. We start by indicating differences in entailments between sentences with ordinary NPs in the direct object position and sentences with proper GNPs in the direct object position. We observe that the former sentences, in contradistinction to the latter, pass the h-test. Consider first the following examples:

(10) a. Leo and Lea hug ten/most students.
b. Bill and Sue hug ten/most students.

(11) Leo, Lea, Bill and Sue hug ten/most students.

It is easy to see that (10a) in conjunction with (10b) entails (11). This is not surprising given the property in (9) and the fact that the VPs in

(10a) and (10b) denote sets. However, sentences with proper GNPs in the object position behave differently in this respect as shown in the following examples:

(12) a. Leo and Lea hug each other/each other and Kim.
b. Bill and Sue hug each other/each other and Kim.

(13) Leo, Lea, Bill and Sue hug each other/each other and Kim.

(14) a. Leo and Lea read the same book/the same five books.
b. Bill and Sue read the same book/the same five books.

(15) Leo, Lea, Bill and Sue read the same books/the same five books.

Clearly, (12a) in conjunction with (12b) does not entail (13). Similarly, (14a) in conjunction with (14b) does not entail (15). In the same way, (12a) and (12b) do not entail that four persons hug each other, and (14a) and (14b) do not entail that four persons read the same book. This means that the functions denoted by the subject NPs in (14a) and (14b) do not apply to the predicate denoted by the complex VPs in these sentences, and the conjunction *and* is not understood pointwise. Furthermore, given property in (9) and proposition 1 this means that the VPs in the above sentences do not denote properties, and that the objects of these sentences do not denote lifts of type $\langle 2 : 1 \rangle$ functions.

Another thing one observes looking at transitive sentences with GNPs as direct object is that they can have virtually any plural NP as their grammatical subject. Thus the following are all acceptable sentences:

(16) Kim and Leo/most students/three teachers/no two monks admire each other.

(17) Between five and ten students/some philosophers read the same book.

In the above sentences GNPs form with the transitive verb a VP, which is a "natural" constituent. Hence, to avoid the type mismatch and get the right interpretations we will consider that the GNPs *each other, each other and Kim*, *the same books* and *the same five books* denote genuine higher order functions on binary relations, that is, functions of type $\langle 2 : \langle 1 \rangle\rangle$.

It is important to keep in mind that there are "many" proper GNPs which have similar behaviour in transitive sentences. For instance all Boolean compounds of *each other* with ordinary NPs or with the reflexive *himself* such as *each other and most teachers* or *each other, themselves and Dan* form such anaphoric GNPs. Similarly, reciprocal determiners (cf.

Zuber 2016) such as *every…except each other* or *most…in addition to each other*, when applied to common nouns, give anaphoric GNPs with similar semantic properties.

There are also "many" comparative GNPs giving rise to similar differences in the entailment. This is the case, for instance, with Boolean compounds such as *the same books and five articles* or *the same five students and one teacher*. In addition, higher order comparative GNPs can be formed with other "comparative" determiners such as *similar, very similar, different, almost the same, almost the same number of, the same kind of, comparable, interchangeable, related, analogous* etc. These determiners can also combine between them in a Boolean style, and the GNPs they form with CNs in their turn can form Boolean compounds. The following examples illustrate some of these possible compounds:

(18) Leo and Dan admire most linguists, except themselves and each other.

(19) Most logicians know the same five and ten different theorems.

(20) No two philosophers admire each other and Plato.

(21) Some students admire each other and the same teachers.

(22) Most Japanese drive very similar cars.

(23) They read the same articles and *Exciting Logic*.

An entailment test similar to the one applied to sentences (14a) and (14b) indicates that the h-test can be applied here to all the above sentences and thus the VPs in these sentences do not denote sets.

I will provide now the semantics for the anaphoric GNPs *each other* and for the comparative GNP *the same CN* using the fact that the VPs they form with transitive verbs are of the category $S/(S/(S/NP))$. The semantics of some other anaphoric GNPs is given in Zuber (2016), and the semantics of some other higher order comparative NPs is given in Zuber (2017). The functions defining the semantics of *each other* and of *the same CN* are important for what follows because they are used to define the semantics of other constructions which induce the VP raising.

Functions corresponding to the semantics of *each other* and *the same CN* use partitions defined above. To define the type $\langle 2 : \langle 1 \rangle \rangle$ function EA denoted by the reciprocal *each other* we use the partition $\Pi(R^{S-})$ (Zuber 2016). This definition is a definition "by cases", which depends on whether the partition $\Pi(R^{S-})$ is trivial or non-trivial. Thus:

D7 (i): $EA(R) = \{Q : Q \in PL \wedge \neg 2(E) \subseteq Q\}$ if $R^{S-} = \emptyset$

(ii): $EA(R) = \{Q : Q \in PL \wedge Q_{D(B)} \subseteq Q\}$, if $\Pi(R^{S-})$ is trivial with B as its only block

(iii): $EA(R) = \{Q : Q \in PL \wedge \exists_B(B \in \Pi(R^{S-}) \wedge Q(D(B) = 1\} \cup \{Q : Q \in PL \wedge \exists_B(B \in \Pi(R^{S-}) \wedge Q = \neg Q_{D(B)}\}$ if $\Pi(R^{S-})$ is non-trivial.

The meaning of *each other*, defined in D7, corresponds to strong logical reciprocity. Weaker reciprocity can be obtained by taking into consideration in D7 some subsets of the relation R^{S-}.

As the second example of a GNP which forces raising of the VP we give the semantics of the comparative GNP *the same CN*. Strictly speaking, we specify the function $SAME(X, R)$, denoted by the (generalised) determiner *the same*. We assume that this determiner denotes a type $\langle 2, 1 : \langle 1 \rangle \rangle$ function. To define this function we use the partition $\Pi_{R_X}(E)$ corresponding to the equivalence relation e_{R_X}, defined in D2 (ii). This again is a definition "by cases". The output of the function to be defined is a set of plural type $\langle 1 \rangle$ quantifiers, which is denoted by the raised VP, will in general contain three parts: positive, negative and "atomic". The positive part corresponds, roughly, to the set of quantifiers true of some block of the partition, and the negative part corresponds to the set of quantifiers that are false of sets which are not blocks of the partition.

We will say that a block of a partition is singular if it is a singleton. A block B is plural, $B \in PL$, if it contains at least two elements. A partition is atomic iff all its blocks are singular. With the help of these notions, using the partition $\Pi_{R_A}(E)$ we can now express the function $SAME(X, R)$, where R is a non-empty binary relation, and X a non-empty set, as follows (Zuber 2017):

D8 $SAME(X, R) =$

(i): $\{Q : Q \in PLR \wedge \neg 2(E) \subseteq Q\}$, if $\Pi_{R_X}(E)$ is atomic

(ii): $\{Q : Q \in PLR \wedge \exists_B(B \in \Pi_{R_X}(E) \wedge B \in PL \wedge Q(B) = 1)\} \cup \{Q : Q \in PLR \wedge \exists_{C \subseteq E} \forall_{B \in \Pi_{R_X}(E)}(C \nsubseteq B \wedge \neg ALL(C) \subseteq Q)\}$, if $\Pi_{R_X}(E)$ is not atomic.

The above definition says that *SAME* applied to a set X and a binary relation R gives as result a set of quantifiers, as desired. This set can be decomposed into various subsets depending indirectly on the "content" of the relation R and thus on the partition of E induced by R and X. According to the clause (i), when the partition is atomic then no two objects are in the relation R with all objects of a subset of X. This entails in particular that the quantifier denoted by *no two objects* and any of its consequences belong to the set $SAME(X, R)$. This means that, for instance, the quantifiers denoted by *no five objects* or *no two students* also belong to the set $SAME(X, R)$.

Clause (ii) concerns the case where the partition is not atomic. In this case there is at least one plural block of the partition such that all its members are, roughly speaking, in the relation R with the same subset of X. This block corresponds to the property expressing the sameness we are looking for and a plural quantifier can be true or false of it. The second part of the clause (ii) provides a set of quantifiers obtained from a "negative information" given by sets which are not blocks of the partition. If, for instance, Jiro and Taro are Japanese students who read different books then no set to which they belong is a block of $\Pi_{R_B}(E)$, where R corresponds to $READ$ and B to $BOOK$. Then, according to the second part of the clause (ii), the quantifiers denoted by the *NPs not all Japanese students*, *not all students* and *not all Japanese* belong to $SAME(B, R)$.

To describe the function denoted by the (generalised) determiner *the same number of* we use the partition corresponding to the equivalence relation $e_{R,n}$ defined in D3(ii) above (cf. Zuber 2017).

Both functions, EA and $SAME$, have specific properties which make them different from any lift of a type $\langle 1 \rangle$ function. Using Proposition 1 it is easy to show that they do not satisfy HEC in particular. Moreover, EA satisfies HPI and $SAME$ satisfies HAI. In addition, these functions have another thing in common: in the description of their content the structure of the relations which are their arguments, in particular the partitions which can be induced by these relations, are explicitly taken into account.

4. Raised intransitive verb phrases

In the preceding section the arguments for raising VPs were based on constructions in which special verbal arguments apply to transitive VPs and give as result raised VPs denoting sets of type $\langle 1 \rangle$ quantifiers. In this section I discuss briefly a somewhat different case of VPs that have to be raised but are not formed from transitive verbs. Here one can distinguish two cases: the case of a raised VP that does not contain any modifier inducing the raising and the case of an intransitive VP that does.

I start with intransitive ("on the surface") verbs that express symmetric relations, such as *to meet* or *to argue* (and not *to meet with* or *to argue with*) and predicates such as *to live on the same street* or *to be an enemy* (and not *to live on the same street as* or *to be an enemy of*). As it has been often noted, subject NPs of sentences with such symmetric predicates have to be interpreted "collectively", since the "property" they express cannot in general apply to individuals, as shown in the following examples :

(24) a. Leo and Lea met (in the park).
b. *Leo met.

(25) a. Most teachers met.
b. *A student met.

(26) ?Leo is an enemy.

On the other hand, sentences with VPs representing symmetric predicates do not pass the h-test: for instance using (24a) and (27) as premisses one cannot obtain (28) as conclusion:

(27) Bill and Dan met.

(28) Leo, Lea, Dan and Bill met.

The verb *to meet* and the predicate *to live on the same street* are interesting in addition for another reason: as indicated above, they are among the predicates that admit implicit or optional arguments. An item which can take a complement is an item with an optional complement if it can occur in a sentence with or without its complement and thus the omission of the complement in a acceptable sentence does not lead to the unacceptability of the sentence, but may lead to some meaning changes. In particular, verbs with optional arguments can occur as intransitive, transitive, or with oblique objects. The verb *to meet* in English, in addition to being intransitive, can take direct and indirect objects. Similarly with other symmetric predicates. In this respect they resemble verbs with GNPs in the form of higher order comparatives:

(29) Leo met Lea.

(30) Leo met with Lea.

(31) Leo read the same book as Lea.

Words with optional complements pose various challenges for formal semantics, one of them being their categorial and lexical ambiguity (Gillon 2012). One can notice that (24a) has two forms logically equivalent to it, with "the same verb" taking either a direct object, as in (29) or an indirect object (in "comitative case"), as in (30). In these sentences with explicit verbal arguments the VPs express a (first order) property and sentences with such VPs and plural NP subjects can have distributive meaning in opposition to the corresponding sentences with omitted verbal arguments.

The semantics of sentences with verbs expressing symmetric relations but in which the complements are omitted necessitates the raising of the type of the verb. Given, however the fact that such sentences are in general equivalent to corresponding sentences with *each other* or *the same* we know already how to compute their meaning. For instance, (27) can be considered as logically equivalent to (32) and (33a) to (33b):

(32) Leo and Lea met each other.

(33) a. Lea and Dan married.
b. Lea and Dan married each other.

As the following examples show not all verbs with implicit complements express symmetric relations:

(34) Leo and Lea undressed.

(35) a. Leo and Lea undressed themselves.
b. Leo and Lea undressed each other.

(36) Leo and Lea kissed.

(37) Leo and Lea kissed each other.

Verbs *undress* and *kiss* are verbs in which arguments are optional and thus they can occur either as intransitive verbs or transitive verbs. Sentence in (34) entails neither (35a) nor (35b), and the one in (36) means, for the pragmatic reasons, only (37). The representations of the "mixed" (reflexive-reciprocal) reading of (34) and of the reciprocal reading of (36) necessitates raising of the intransitive verbs *undress* and *kiss*.

One of the consequences of the above observations is that verbs admitting omitted arguments can take their denotations in three different denotational algebras: in $D_{(S/NP)/NP}$, in D_{VP} and in D_{VP^R}. This situation is similar to the one finds with some NPs, which can also denote in three different types (Partee 1987).

Let us see now some examples of constructions where raising is induced by some verbal modifiers (that is adverbial phrases) and not by verbal arguments. Before introducing adverbs that force the raising of VPs, it is important to observe that they do not belong to the class of "classical" adverbs of quantification with non-nominal domain forming adverbials or prepositional phrases. For instance (non-nominal) quantifiers such as *always, everywhere, never, nowhere, often, most of the time, on some occasions* etc. do not force VP raising. Sentences with these adverbs do pass the h-test: (38) and (39) together entail (40):

(38) Dan never drinks.

(39) Most monks never drink.

(40) Dan and most monks never drink.

A good candidate for an adverb forcing the raising of the VP is the adverb *together*. Detailed semantics of this adverb may involve various aspects (cf. Moltmann 2004) that will not be discussed here. Consider the following examples:

(41) a. Kim and Dan left together.
b. Leo and Lea left together.

(42) Kim, Dan, Leo and Lea left together.

One observes that the above sentences behave like transitive sentences with GNPs and sentences with omitted verbal arguments. Thus, (41a) in conjunction with (41b) does not entail (42). This means, according to Proposition 1 that the type of the object denoted by the VP *left together* is different from $\langle e, t \rangle$.

Sentences with VPs modified by the adverb *together* can also take as subject virtually any plural NP, in the same way as transitive sentences can take a proper GNP in the direct object position:

(43) Some/most/ten students/Leo and Kim left together.

It is worth recalling that in many languages the "reciprocal morpheme", which gives rise to proper GNPs with the reciprocal meaning we discussed above, can have many uses and carry multiple "meanings" (Lichtenberk 1985). In particular, in languages related to Turkish this morpheme can carry the so-called social or associative meaning expressed in English by the adverbial *together*. So it should not be surprising that there are adverbials forcing type raising of VPs even if they are categorially different from nominal verbal arguments discussed in the preceding section.

For present purposes it is enough to notice that (41a) can be considered as equivalent to (44):

(44) Kim and Dan left with each other.

In this case *left with* can be considered as expressing a binary relation and thus the raising of the VP is necessary because of the presence of the GNP *each other*.

The situation is probably more complicated in (43). Very likely in this case we need a "weaker" *together*: it is not necessary that any member of the group of ten students or of the group representing the majority of students leaves with every other member of the group. In other words *together* in (43) should be defined by a weaker *each other*.

An adverb related to *together* is the adverb *separately*. One can check that sentences with VPs modified by this adverb do not pass the h-test and thus this adverb also induces VP-raising. Similarly, adverbs related to *the same* such as *in the same way, equally, differently*, etc. induce VP raising. Thus, to get the semantics of the VPs such as *argue in the same way, be equally stupid* and *dress differently* we have to raise their type.

Gerundives in many languages can act as VP modifiers, as for instance in *to dance singing and laughing* or *to sit reading a book*. It seems natural to consider that gerundives used as modifiers of VPs and formed from raised VP force the raising of the VP which they modify:

(45) Leo and Lea came using the same taxi.

(46) Lea and Dan left kissing each other.

To obtain the semantics of the above sentences the VPs have to be raised. I will not show this in detail since, in particular, it involves the semantics of gerundives in general. It suffices to notice that in many cases gerundival adverbials can be expressed by the conjunction of the modified VP with the one from which the gerundive is formed. For instance, (47) can be considered as being logically equivalent to (48):

(47) Lea and Dan were dancing talking to each other.

(48) Lea and Dan were dancing and talking to each other.

Recall that one of the arguments for NP raising is based on Boolean compounds. This argument is related to the use of proper nouns in Boolean compounds with quantified NPs: roughly speaking, in order to compute the meaning of such compounds all members of the compound have to denote in the same type and thus the type of the proper nouns has to be lifted from e to $\langle\langle e,t\rangle,t\rangle$. The same argument applies in the case of the VP raising: one cannot conjoin, for instance, a raised VP and a non-raised one if one wants to compute the meaning of the whole conjunction.

The argument for VP raising based on Boolean compounds applies not only to cases with gerundival modification. Consider the following examples:

(49) Leo and Lea left and took the same taxi.

(50) Most students danced, sang the same song and held each other's hands.

(51) Some monks met and discussed jokes.

Although the semantics of the first VPs in (49) and in (50) can be given without raising them when they are in isolation, being conjoined with raised VPs in these sentences they too must be raised. Similarly, in (51) the VP *discussed politics* must be raised since it occurs in a conjunction with the raised VP *met*.

The fact that some adverbs inducing the raising can be "described" with the help of GNPs such as *each other* and *the same* allows us to see in a different light some hard problems related to the semantics of cumulative readings of some quantifiers in specific contexts. When one thinks about the famous example of piano lifters (as in (52a)), it becomes obvious that the cumulative reading entails that the lifters lifted the piano with *each other* and that it was *the same* piano. In fact, strictly speaking, *the same* in case is weaker than *the same* defined in D8 because it only inverses the scope of the direct object (Zuber 2017). Thus the meaning of (52a), with the cumulative interpretation of its subject NP, can be expressed by something like (52b). Similarly, (53a) can be paraphrased by (53b):

(52) a. Leo and Dan lifted the piano.
b. There is a piano such that Leo and Dan lifted it together.

(53) a. Three philosophers wrote nine articles (for the journal).
b. There are nine articles (of the journal) such that three philosophers wrote them together.

The presence of the modifier *together*, taken possibly in its weak reading, in the above sentences is essential. In general both, subject and object, NPs in cumulative readings are scopeless, but in this case the presence of *together* allows for a representation with the object NP taking wide scope.

In fact, to have cumulative/collective readings, both the adverbial *together* and the comparative *the same* have to occur: (54a) does not and (54b) does express a collective/cumulative action:

(54) a. Leo and Dan read the same book.
b. Leo and Dan read the same book together.

The example in (54b) shows that functions forcing VP raising may be predicate and argument invariant "at the same time".

I conclude this section by indicating that the so-called categorially polyvalent modifiers such as *only, even, also*, etc. can also be considered as inducing VP raising when they have intransitive VPs in their scope. This point will not be developed here.

5. Conclusive remarks

One of the most often used applications of type raising is related to the difficulty of dealing with the semantics of "plural" NPs. In fact one can notice that even "simple" sentences whose subject NP is a conjunction of two proper nouns, and the VP is marked by the plural verbal marker, do not pass the h-test. For this reason, many operators defining specific raisings of NPs, or even of the (nominal) determiner forming a NP, have been proposed. In this paper I argue for the usefulness of the "classical raising" strictly related to Montague's NP raising, without any additional "non-classical" raising of determiners. It can be defined by set-theoretical (type theoretical) means. Such VP raising is necessary for the compositional semantics of various complex predicates whose readings are difficult, if not impossible, to express in first order logic.

No formal results concerning VP raising have been presented. At least two kinds of questions related to formal properties have to be investigated. The first concerns the constraints that should be imposed on the content of raised VPs and on the operation leading to the raising. We have seen that sets of quantifiers denoted by properly raised VPs are not verbal individuals because they are closed with respect to meets. It seems, however, that any set of quantifiers denoted by a properly raised VP is increasing in the sense that if a quantifier Q_1 belongs to it and $Q_1 \subseteq Q_2$ then Q_2 also belongs to it. For instance we see that (13) above entails (12a) and (12b). Similarly, (15) entails (14a) and (14b).

All examples we have discussed essentially involve the plurality of the subject NPs in sentences with a raised VP. It seems thus obvious that individuals should be in some way excluded from the set of quantifiers corresponding to a raised VP and thus the constraint on raising should take into account the particular status of individuals. The set of quantifiers denoted by properly raised VPs should also be consistent. Less obvious is the constraint of completeness. We have seen that raised predicates involve plurality and so probably nominal individuals should be excluded from their denotations in some way. However, it is not clear whether such plural predicates with singular subject NPs should be considered as non-grammatical or just give rise to false sentences.

The second point concerns the status of other operations that go together with the classical raising, like for instance the operation of *lowering* a raised VP. For instance, we need to know when, if ever, and why a raised VP can be lowered in order to get its primitive type $\langle e,t\rangle$. More specifically we want to know under what conditions to a given set $\diamond VP$ of type $\langle 1\rangle$ quantifiers one can associate by an operation, that is the inverse to the VP raising, a set (of individuals) such that by raising this set we get the given set $\diamond VP$ of quantifiers. Recall that in the case of the "classical" NP raising the corresponding inverse operation is a mapping LOW from type $\langle 1\rangle$ quantifiers to elements of E. More precisely, it is a partial mapping that applies to nominal individuals, treated as quantifiers (principal ultrafilters) and maps such quantifiers to their generators (Partee 1987). The situation is quite similar in the case of VP raising: any set of type $\langle 1\rangle$ quantifiers that is a verbal individual can be lowered to a set. This set is obtained by taking the meet of all nominal individual members of the given verbal individual. Of course, only sets of quantifiers that are verbal individuals can be lowered in this way. For instance for any binary relation R and any type $\langle 2:1\rangle$ function F, the set $F^L(R)$ of type $\langle 1\rangle$ quantifiers (where F^L is defined as in (7)) can be lowered: $LOW(F^L(R)) = F(R)$.

Another series of questions related to the VP raising concerns its complexity and possible strategies for processing sentences with raised VPs. Van Benthem (1986) proposes to measure the semantic complexity of types by the function o of order which assigns to any type a natural number. It is defined recursively as follows:

(55) a. $o(e) = o(t) = 1$

b. $o(\langle a,b\rangle) = max(o(a)+1, o(b))$

Given this measure the complexity the type of raised VPs is of order 3. This order is not higher than the order of the type of (nominal) determiners (type $\langle\langle e,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle$) or the type of prepositions (type $\langle\langle\langle e,t\rangle t\rangle,\langle\langle e,t\rangle,\langle e,t\rangle\rangle$). Van Benthem indicates that order 3 is sometimes considered as the threshold for natural languages. Given the fact that the order of raised VPs is 3, one can consider that the operation of VP raising does not go beyond this threshold. It is not clear, however, what the consequences of this fact are for the way sentences with raised VPs can be processed.

References

Bach, E. 1980. Tenses and aspects as functions on verb phrases. In C. Rohrer (ed.) Time, tense and aspect. Tübingen: Niemeyer. 19–37.

Barwise, J. and R. Cooper. 1981. Generalized quantifiers and natural language. Linguistics and Philosophy 4. 159–219.

Benthem, J., van. 1986. Essays in logical semantics. Dordrecht: Reidel.

Does, J., van der. 1993. Sums and quantifiers. Linguistics and Philosophy 16. 509–550.

Geach, P. 1972. A program for syntax. In D. Davidson and G. Harman (eds.) Semantics of natural language. Dordrecht: Reidel. 483–497.

Gillon, B. S. 2012. Mass terms. Philosophy Compass 7. 712–730.

Keenan, E. 2016. *In situ* interpretation without type mismatches. Journal of Semantics 33. 87–106.

Keenan, E. and L. Faltz. 1985. Boolean semantics for natural language. Dordrecht: Reidel.

Lichtenberk, F. 1985. Multiple uses of reciprocal constructions. Australian Journal of Linguistics 5. 19–41.

Moltmann, F. 2004. The semantics of *together*. Natural Language Semantics 12. 289–318.

Partee, B. H. 1987. Noun phrase interpretation and type-shifting principles. In J. Groenendijk, D. de Jongh and M. Stokhof (eds.) Studies in discourse representation theory and the theory of generalized quantifiers. Dordrecht: Foris. 115–143.

Partee, B. H. and M. Rooth. 1983. Generalized conjunction and type ambiguity. In R. Bäuerle, C. Schwarze and A. von Stechow (eds.) Meaning, use and the interpretation of language. Berlin & New York: de Gruyter. 361–383.

Zuber, R. 2014. Generalising predicate and argument invariance. In N. Asher and S. Soloviev (eds.) Logical Aspects of Computatioonal Linguistics, LACL 2014 Proceedings. Berlin & New York: Springer. 163–176.

Zuber, R. 2016. Anaphors and quantifiers. In J. Väänänen, Å. Hirvonen and R. de Queiroz (eds.) Logic, language, information, and computation (23rd International Workshop, WoLLIC 2016, Puebla, Mexico, August 16–19th, 2016. Proceedings). Berlin: Springer. 432–445.

Zuber, R. 2018. On generalized noun phrases. In A. Foret, R. Muskens and S. Pogodalla (eds.) Formal Grammar 2017. Berlin & New York: Springer. 142–155.