Journal of Language Modelling

VOLUME 3 ISSUE 2
DECEMBER 2015

*Institute of Computer Science
Polish Academy of Sciences
Warsaw*

Journal of Language Modelling

VOLUME 3 ISSUE 2
DECEMBER 2015

JOURNAL OF
LANGUAGE MODELLING

ISSN 2299-8470 (electronic version)
ISSN 2299-856X (printed version)
`http://jlm.ipipan.waw.pl/`

Published by IPI PAN
Instytut Podstaw Informatyki
Polskiej Akademii Nauk
Institute of Computer Science
Polish Academy of Sciences
ul. Jana Kazimierza 5, 01-248 Warszawa, Poland

Circulation: 100 + print on demand

Layout designed by Adam Twardoch.
Typeset in XeLaTeX using the typefaces: *Playfair Display*
by Claus Eggers Sørensen, *Charis SIL* by SIL International,
JLM monogram by Łukasz Dziedzic.

EDITORIAL BOARD

Predicting word order universals

Paola Merlo
University of Geneva, Department of Linguistics,
Geneva, Switzerland

ABSTRACT

This paper shows a computational learning paradigm to compare and test theories about language universals. Its main contribution lies in the illustration of the encoding and comparison of theories about typological universals to measure the generalisation ability of these theories. In so doing, this method uncovers hidden dependencies between theoretical dimensions and primitives that were considered independent and independently motivated.

Keywords: frequency, word order universals, Greenberg's Universal 20, computational modelling, probabilistic models, Naive Bayes

1 MULTILINGUAL COMPUTATIONAL MODELLING OF LANGUAGE

Current computational linguistic work shows great interest in extending successful probabilistic modelling to multilingual approaches. Many tasks and applications, such as tagging or parsing, are being investigated in a multilingual perspective. The final goal of this line of work is to uncover cross-linguistic regularities to automatically extend new techniques and technologies to new languages, and to make use of large amounts of data.

Computational modelling can interact with large-scale linguistic work at other interesting levels. From the point of view of the theory, the properties of the computational models might shed light on some of the properties of the generative processes underlying natural language. From the point of view of the data, computational models can be used to develop and test correlations between different aspects of the data on a large scale. Methodologically, computational models and machine learning techniques provide robust tools to test the predictive power of the proposed generalisation.

Language universals – whether defined as linguistic properties, observed or very abstract, that are exhibited by all languages or as statistical implications of pairs of linguistic properties – are at the moment a topic of great debate. Their nature and even their existence has been called into question (Dunn *et al.* 2011) and their general nature and distribution are being investigated from a formal and cognitive point of view (Cinque 2005; Cysouw 2010a; Steedman 2011; Culbertson *et al.* 2012; Culbertson and Smolensky 2012; Futrell *et al.* 2015).

We will specifically concentrate on the quantitative properties of word order universals (Dryer 1992; Cysouw 2010b; Steedman 2011). In this debate, it is of great interest to attempt to explain not only the possible or impossible word orders as attested by typological traditions, but also their distribution. Data-driven computational models can help cast light on this question in two main ways. First, through their formal nature, they can make the assumptions in the proposals explicit and operational. Second, through the large-scale that is inherently possible with automatic methods, claims can be quantified and verified not only at the level of language type, but also at the level of linguistic token, for each individual language.

This paper concentrates on a central methodological point. It will illustrate how to formalise some of the current proposals for the much debated Universal 20 (Greenberg 1966) – the universal governing the linear order of a noun and its modifiers – in such a way that they can be evaluated and compared quantitatively in a setting where their ability to generalise to new cases is properly tested. In this respect, this work shares the goals of Cysouw (2010a), but differently from these previous proposals of the same nature, the proposed theories are encoded as faithfully as possible, by using their defined primitives and operations as features in our models.

2 THE FACTS

One of the most easily observable distinguishing features of human languages is the order of words: the order of the main grammatical functions in the sentence, the position of the verb in the sentence, and the respective order of the modifiers of a noun, among others.

While there is great variety in the orders, most languages have very strong preferences for a few or only one order, and, across languages, not all orders are equally preferred (Greenberg 1966; Dryer 1992). Greenberg's universal 20 describes the cross-linguistic preferences for the word order of elements inside the noun phrase.

> **Greenberg's Universal 20**
> When any or all the items (demonstrative, numeral, and descriptive adjective) precede the noun, they are always found in this order. If they follow, the order is exactly the same or its exact opposite.

We can reformulate universal 20 more explicitly (Cinque 2005):

(a) In prenominal position, the order of demonstrative, numeral, and adjective is Dem>Num>A.

(b) In postnominal position, the order is either Dem>Num>A or A>Num>Dem.

Some aspects of Greenberg's formulation have withstood the test of time, but some others have been found to be too strong. (See, for example, Dryer's and Cinque's large data collections in the cited work.) On the one hand, a larger sample of languages has shown that two of the three orders indicated by Greenberg's as the only possible orders are indeed among the most frequent ones. On the other hand, larger samples have also shown that many more orders are possible than stated in Greenberg's universal, but with different frequencies (Cinque 2005; Dryer 2006).

Establishing the actual basic facts is not so simple. We will concentrate here only on the quantitative aspects and will assume without argument the results described in the literature that assign certain languages to certain word orders. In assessing the reliability of the proposed counts, one has to assess the possible sources of errors induced by sampling. Sampling, in general, is subject to random error and to bias error. Random error occurs when the size of the sample is not adequate to the complexity of the problem, so that some possible events are not observed. Greenberg's sample of languages was probably too small, and inspections of larger samples have discovered some orders that looked impossible.

Bias error occurs when the nature of the sample is biased with respect to the conclusions one wants to draw. To draw conclusions on language universals, it is therefore crucial that the sample be representative of the true underlying linguistic diversity, for example, as generated by a posited probabilistic system. The remedy to random error is to have a sufficiently large number of data points: Dryer's and Cinque's current language collections range in the hundreds. To address the problem of bias error, Dryer suggests counting language genera and not individual languages, since some genera are much more densely populated, and better studied, than others (Dryer 2006).[1]

Table 1 reports the 24 combinatorially possible orders of the four elements: N, Dem, Num, Adj and the actual counts that have been proposed in several publications: the first column shows discretised frequencies; the following two columns are Dryer's (2006) counts by language and by genera; and the following column are Cinque's counts, as can be deduced from the 2005 paper. In the first column, the discretised frequencies are calculated according to Dryer's counts of genera. As can be observed, there are some discrepancies across the different counting methods and across authors, which have been discussed in detail in the related publications, but also many points of agreement. In particular, while the exact numbers sometimes vary, the rank of languages or genera based on frequencies is almost identical. This observation indicates that aiming to predict the frequency rank, as opposed to exact frequency counts, would be more robust across theories and more robust to new observations. The numerical frequency data are then transformed into ordered data by a process of discretisation and then used by a discrete classifier. The discretisations can be done at different levels of granularity. Table 2 shows a two-way, four-way, and seven-way discretisation. More will be said about this discretisation later. In what follows, therefore, we investigate how different theories fare in explaining different levels of frequency of word orders and how well they generalise this prediction to previously unseen data.

[1] Dryer (2005, 584) provides the following definition: "A genus is a group of languages whose relatedness is fairly obvious without systematic comparative analysis and which even the most conservative "splitter" would accept.". (An explanation of genus is also available on WALS online at `http://wals.info/languoid/genealogy`.) Examples are such subfamilies of Indo-European as Germanic, Slavic, and Romance languages.

Table 1: Attested word orders of Universal 20 and their estimated frequencies. (See the text for more explanation.)

				D’s	D’s	D’s	C’s
				Discr	Lang	Gen	Freq
Dem	Num	Adj	N	V. Freq	74	44	V. many†
Dem	Adj	Num	N	Rare	3	2	0
Num	Dem	Adj	N	0	0	0	0
Num	Adj	Dem	N	0	0	0	0
Adj	Dem	Num	N	0	0	0	0
Adj	Num	Dem	N	0	0	0	0
Dem	Num	N	Adj	Freq	22	17	Many*
Dem	Adj	N	Num	Rare	11	6	V. few (7)
Num	Dem	N	Adj	0	0	0	0
Num	Adj	N	Dem	Rare	4	3	V. few (8)
Adj	Dem	N	Num	0	0	0	0
Adj	Num	N	Dem	0	0	0	0
Dem	N	Adj	Num	Freq	28	22	Many**
Dem	N	Num	Adj	Rare	3	3	V. few (4)
Num	N	Dem	Adj	Rare	5	3	0
Num	N	Adj	Dem	Freq	38	21	Few (2)
Adj	N	Dem	Num	Rare	4	2	V. few (3)
Adj	N	Num	Dem	Rare	2	1	V. few
N	Dem	Num	Adj	Rare	4	3	Few (8)
N	Dem	Adj	Num	Rare	6	4	V. few (3)
N	Num	Dem	Adj	Rare	1	1	0
N	Num	Adj	Dem	Rare	9	7	Few (7)
N	Adj	Dem	Num	Freq	19	11	Few (8)
N	Adj	Num	Dem	V. Freq	108	57	V. many (27)

† The exact counts are not provided.

* Cinque mentions European languages and 13 others.

** Ten languages and alternative order for three more.

Table 2: Two-way (possible or 0), four-way (very frequent, frequent, rare, none, abbreviated as VF,F,R,0) and seven-way (57,44,22,11,6,3,0) discretisation and the observed counts based on genera from Dryer's.

				Two-way Discr	Four-way Discr	Seven-way Discr	Dryer's Genera
Dem	Num	Adj	N	Possible	VF	44	44
Dem	Adj	Num	N	Possible	R	3	2
Num	Dem	Adj	N	0	0	0	0
Num	Adj	Dem	N	0	0	0	0
Adj	Dem	Num	N	0	0	0	0
Adj	Num	Dem	N	0	0	0	0
Dem	Num	N	Adj	Possible	F	22	17
Dem	Adj	N	Num	Possible	R	6	6
Num	Dem	N	Adj	0	0	0	0
Num	Adj	N	Dem	Possible	R	3	3
Adj	Dem	N	Num	0	0	0	0
Adj	Num	N	Dem	0	0	0	0
Dem	N	Adj	Num	Possible	F	22	22
Dem	N	Num	Adj	Possible	R	3	3
Num	N	Dem	Adj	Possible	R	3	3
Num	N	Adj	Dem	Possible	F	22	21
Adj	N	Dem	Num	Possible	R	3	2
Adj	N	Num	Dem	Possible	R	3	1
N	Dem	Num	Adj	Possible	R	3	3
N	Dem	Adj	Num	Possible	R	3	4
N	Num	Dem	Adj	Possible	R	3	1
N	Num	Adj	Dem	Possible	R	6	7
N	Adj	Dem	Num	Possible	F	11	11
N	Adj	Num	Dem	Possible	VF	57	57

3 SOME THEORIES

We will compare the descriptive and predictive adequacy of a few of the proposals that have been put forth to explain Greenberg's Universal 20, choosing a few theories that have different properties.

In a paper that has received much commentary (Cinque 2005), Greenberg's Universal 20 is derived from independently motivated principles of syntax organised in a derivational explanation. Based on data as those shown in the fifth column of Table 1, Cinque remarks that there are 24 combinatorially possible orders of the four elements: N, Dem, Num, Adj. According to Cinque, only 14 of them are attested in the languages of the world (but see Dryer's counts in the same table, Table 1). Some of the 14 orders are unexpected under Universal 20. Cinque proposes that the actually attested orders, and none of the unattested ones, are derivable from a single universal order of the basic constructive syntactic operator (the Linear Correspondence Axiom, Kayne 1994), and from independent conditions on phrasal movement. The Linear Correspondence Axiom first combines Nouns and Adjectives, then adds Numerals and finally adds Demonstratives. Different types of movement can move the merged elements to different positions in the phrase: all the way to the beginning of the phrase or only partially. These conditions enable one to consider some forms of movement as more costly than others and no movement as the preferred unmarked option. In this way, Cinque's proposal also derives the exceptions, and the different degrees of markedness of the various orders.

In a different proposal, a factorial, but not derivational, explanation is proposed (Cysouw 2010a). Statistical models are used and an explanation of typological frequencies is produced by the cumulative combination of various interacting characteristics. The author experiments with various models to see which one better predicts the attested frequencies. Three characteristics are used by all models of the NP-internal word order: hierarchical structure, noun-adjective order, and whether the noun is at the phrase boundary. In a further simplification of the model, the hierarchical structure can be broken down into less complex features (noun-adjective co-occurrence, demonstrative at the edge of the phrase, and noun at the edge of the phrase).[2]

[2] Like Cinque, Cysouw is concerned with demonstrating that the proposed

This factorial explanation does not provide a generative process that explains how the different word orders could arise from a common grammar, but it identifies the predictive properties of the frequency distributions of word order and their relative importance.

Dryer proposes a factorial explanation based on general principles of symmetry and harmony (Dryer 2006). Differently from Cinque's and Cysouw's, this proposal does not assign any weights to the factors. The factors comprise two symmetry principles that describe the closeness of the modifiers to the noun; a principle of asymmetry that captures the main observation that prenominal modifiers exhibit fewer alternatives than post-nominal modifiers (also observed by Cinque); a principle of intra-categorial harmony; and universal 18. Figure 1 spells out the principles. What is really very important in Dryer's contribution are the provided observed frequencies. On the one hand, Dryer shows that a few of the word orders that Cinque had declared impossible are actually attested, one of them quite frequently. On the other hand, it provides frequency counts based on genera and not simply on languages, based on an independently justified sampling procedure that factors out influences of language family. These genus-based counts are used in our study, and are shown in Table 1.

In conclusion, all these theories attempt to describe the very different frequency counts of types of languages by proposing factors that favour harmonic orders, and that derive the asymmetry between

principles are not limited to explaining Universal 20. To strengthen the generality of the proposed method, Cysouw discusses how it can also be used to explain the typology of sentence word order, as it is captured by Greenberg's Universal 1. Recall Universal 1: "In declarative sentences with nominal subject and object, the dominant order is almost always one in which the subject precedes the object." This universal holds for 96% of the world's languages, but it does not model the finer-grained differences in frequency of the six word order types. Cysouw proposes a more complex three-feature model. The first feature is pairwise order: whether the order is SO or OS, VO or OV, SV or VS. The second feature is pairwise adjacency: for instance, whether S and O are adjacent or not. The third feature is individual position: for instance, whether S is first, medial, or final. Cysouw shows that the first two features are less important than the third and that overall the model has a better fit than universal 1. However, notice that this model comprises two three-valued features and one binary feature, so it has five degrees of freedom. These are enough degrees of freedom to simply list all the six possible word orders of the three S,O,V elements.

- **Symmetry Principle 1**
  The adjective and numeral tend to occur closer to the noun than the determiner, when they occur on the same side of the noun.
- **Symmetry Principle 2**
  The adjective tends to occur closer to the noun than the numeral, when they occur on the same side of the noun.
- **Asymmetry Principle**
  The symmetry principles tend to apply more strongly to prenominal modifiers than to postnominal modifiers; exceptions to the symmetry principles will occur only to the postnominal modifiers.
- **Greenberg's Universal 18**
  When the descriptive adjective precedes the noun, the demonstrative and the numeral, with overwhelmingly more than chance frequency do likewise.
- **Intra-categorial Harmony**
  The demonstrative, numeral, and adjective tend to all occurr on the same side of the noun.

Figure 1: The five principles used in Dryer's explanation of Universal 20.

prenominal and post-nominal modifiers. They all try to fit the frequency distribution of the languages to the models and to compare to other proposals. In the rest of the paper, we illustrate an encoding and an automatic learning method to test how well these models predict the observed distributions of word orders.

4 BUILDING PREDICTIVE MODELS

In this section, we test the generalising ability of some of the different explanations that have been proposed for Universal 20. The method will require transforming the three theories into a vectorial representation, as described below, and then automatically finding the relative weight of each element in the vector, a process of parameter fitting. We use the ability to classify new instances in a supervised learning setting as an indication of the generalising power of the theory. We compare the three theories described above.

Fitting parameters to a model based on available data gives us a measure of the descriptive fit of the model to the data, an interesting measure in itself, but it does not test the power of generalisation of the model. This is because it is always possible to fit the data if the number of parameters in the model is sufficiently large given the amount of variation to explain. (For a similar point with a different example, see also Abney 2011.) So the true test of generalisation of a model cannot lie in showing that all the data is explained if that data was actually used to determine some aspects of the model. In explaining Universal 20, what needs to be shown is that the same set of operations and (markedness) weights that capture the observed data also predicts new data to a good degree. In practice, the proper procedure requires fitting the weights on a subset of languages (the training data), and seeing if the quantitative model so developed predicts the frequency distribution for test data not seen during training.

The steps of the simple formalisation that we propose here, therefore, are as follows:

1. Formalise the properties and operations of a model of word order as simple primitive features with a set of associated values;
2. Encode each word order as a vector of instantiated primitives defined by the model;
3. Learn the model through a learning algorithm on a subset of the data;
4. Run the model on previously unseen data to test generalisation ability.

In the rest of the section, we briefly illustrate the feature-based formalisation of the linguistic proposals, and describe the experimental materials and method.

4.1 *Materials*

The different linguistic proposals are translated into a feature-based summary description of each of the word orders. This vectorial representation of the data is compatible with many different training regimes and algorithms. Two proposals (Cysouw's and Dryer's) are declarative, and therefore easily transferred in the simple declarative feature-based framework. Cinque's model is derivational and requires the most interpretation to be formalised and translated into features. We describe here the process to reach this conversion in detail.

In the simplest set up, we code the principles and operations proposed by Cinque for each word order as a vector of properties, a summary that describes each language and its word order. To explain the frequency distribution of the word orders, Cinque affects markedness weights to the different types of move operations. In the computational terminology that will be used below, these weights are the parameters of Cinque's model, and this process is a process of parameter fitting on the available data.

Recall that the salient property of Cinque's explanation is the interaction between a fixed universal word order (the Linear Correspondence Axiom) and structure movement operations, with different markedness weights. A simplified specification of Cinque's explanation for each word order can be encoded as the values of three merge operations and the values of two types of move operations, partial and complete movement. The three merge operations build the structure linearly, corresponding to the word order. Some word orders that require merge sequences not allowed by the Linear Correspondence Axiom are encoded as negative data. The move operations can move elements one step, two steps, that is they can be partial movement, or all the way to the beginning of the phrase, as complete movement. These two types of move operations can be of several kinds, NP-movement, pied-piping, among others. It is crucial to point out that this is only a *model* of Cinque's explanation, limited only to the discriminating features. For example, the fact that there are two movement types in the description of each word order does not imply that there are necessarily two movement steps. There could be more than one partial movement or none. In the vectorial representation, all partial movements (that is, movements that do not reach the left edge of the phrase) are reduced to one value.

The features and the possible values of Cinque's model are shown in Figure 2. *First, second* and *third* represent the three merge operations, and their values are the pairs of syntactic part-of-speech-tags of heads that are being merged (we assume a dependency representation for the trees). *Partial* and *complete* are the two features representing the two movements, and their possible values, which encode the types of movement that Cinque postulates. The values of this feature are *not*, encoding the fact that no movement has taken place, *np*, encoding the movement of the NP alone, *of-who-pp*, encoding NP-movement with pied-piping of the *picture of who* type, and

Figure 2: Cinque's move and merge feature vectors. (See the text for explanation.)

- Template: < first, second, third, partial, complete, frequency >
- Attributes and Values
  - first: AN, DN, ND, NNum, NumN
  - second: AD, DA, DNum, NumA, NumD, NumN
  - third: AD, AN, ANum, DNum, NumA, NumD
  - partial: not, np, of-who-pp, whose-pp
  - complete: not, np, of-who-pp, whose-pp
  - frequency: very frequent, frequent, rare, none (VF,F,R,No)
- Vectors

AN	NumA	DNum	not	not	VF
NumN	DNum	AN	not	not	R
AN	DA	NumD	not	not	No
DN	AD	NumA	not	not	No
NumN	DNum	AD	not	not	No
DN	NumD	ANum	not	not	No
AN	NumA	DNum	whose-pp	not	F
AN	NumA	DNum	of-who-pp	not	R
AN	DA	NumD	whose-pp	not	No
AN	NumA	DNum	not	of-who-pp	R
NNum	DNum	AD	not	not	No
ND	NumN	ANum	not	not	No
AN	NumA	DNum	whose-pp	not	F
AN	NumA	DNum	np	not	R
AN	DA	NumD	not	not	R
AN	NumA	DNum	np	of-who-pp	F
AN	NumA	DNum	not	of-who-pp	R
AN	NumA	DNum	of-who-pp	whose-pp	R
AN	NumA	DNum	np	not	R
AN	NumA	DNum	whose-pp	np	R
AN	NumA	DNum	np	np	R
AN	NumA	DNum	np	whose-pp	R
AN	NumA	DNum	whose-pp	np	F
AN	NumA	DNum	whose-pp	whose-pp	VF

whose-pp, encoding NP-movement with pied-piping of the *whose picture* type.[3]

The values in the last column are the frequency property of the word order, the dependent variable we are trying to explain. We discuss them below.

Recall that Cysouw proposes a factorial explanation, where factors are preferences of directionality and surface proximity. Cysouw shows that three factors are sufficient to achieve a good fit to the data, and argues that a model with fewer parameters should be preferred to a model with more parameters: whether the Noun is near the edge of the Noun phrase or not, whether the Demonstrative is near the edge or not, and whether the Adjective is near the Noun. These are surface observed properties that can be encoded directly in the vector of features that describes each word order. The resulting features, feature values, and vectors are shown in Figure 3.

Dryer's factorial explanation is based on general principles of symmetry and harmony, and does not use any weighing coefficients. Again, these are observed properties that can be encoded directly in the vector of features that describes each word order. The resulting features, feature values, and vectors are shown in Figure 4.

The goal attribute, the attribute we are trying to predict, is the frequency of a given word order. Since the actual counts of languages are still under discussion, and therefore are not entirely reliable, it is a better representation of the current state of reliability of the frequency counts to group them in frequency classes. We can group the languages in different frequency groups, by discretising the frequencies in different ways: either as simply possible or impossible (two values), or as having different levels of frequency. Table 2 shows the different discretisaton values and how they compare to Dryer's counts based on genera. We defined four and seven discrete values, based on observation of the groupings of the actual numerical values. Figures 2, 3,

[3] Many instances of wh-movement involve pied-piping. Pied-piping occurs when a fronted wh-word pulls an entire encompassing phrase to the front of the clause. Cinque indicates that *picture of who* pied-piped movement moves a cluster of the form [XP[NP]], while the *whose picture* type moves [NP[XP]]. The names refer to the two constructions in questions such as *Whose pictures are you looking at?* and relative clauses such as *Mary, your picture of whom/whose picture Tom likes, is very nice.*

Figure 3: Cysouw's feature vectors. (See the text for explanation.)

- Template: < NA-adjacency, N-edge, Dem-edge, frequency >
- Attributes and Values
  - NA-adjacency: Y, N
  - N-edge: Y, N
  - Dem-edge: Y, N
  - frequency: very frequent, frequent, rare, none (VF,F,R,No)
- Vectors

Y	Y	Y	VF
Y	Y	Y	R
Y	Y	N	No
N	Y	N	No
N	Y	N	No
N	Y	N	No
Y	N	Y	F
Y	N	Y	R
Y	N	N	No
Y	N	N	R
N	N	N	No
N	N	N	No
Y	N	Y	F
N	N	Y	R
N	N	N	R
Y	N	N	F
Y	N	N	R
Y	N	N	R
N	Y	N	R
N	Y	N	R
N	Y	N	R
N	Y	Y	R
Y	Y	N	F
Y	Y	Y	VF

- Template:
  <symmetry1, symmetry2, asymmetry, U18, harmony, frequency>
- Attributes and Values
  - symmetry1: Y, N
  - symmetry2: Y, N
  - asymmetry: Y, N
  - U18: Y, N
  - harmony: Y, N
  - frequency: very frequent, frequent, rare, none (VF,F,R,No)
- Vectors

Y	Y	Y	Y	Y	VF
Y	N	N	Y	Y	R
N	Y	N	Y	Y	No
N	Y	N	Y	Y	No
N	N	N	Y	Y	No
N	N	N	Y	Y	No
Y	Y	Y	Y	N	F
Y	Y	Y	N	N	R
N	Y	N	Y	N	No
Y	Y	Y	N	N	R
N	Y	N	N	N	No
Y	N	N	N	N	No
Y	Y	Y	Y	N	F
Y	N	Y	Y	N	R
N	Y	Y	Y	N	R
Y	Y	Y	Y	N	F
N	Y	Y	N	N	R
Y	Y	Y	N	N	R
N	N	Y	Y	Y	R
N	Y	Y	Y	Y	R
N	N	Y	Y	Y	R
Y	N	Y	Y	Y	R
N	Y	Y	Y	Y	F
Y	Y	Y	Y	Y	VF

Figure 4: Dryer's feature vectors. (See the text for explanation.)

and 4 show a four-way discretisation into very frequent (VF), frequent (F), rare (R), and unattested (No). Notice that the fact that we also encode unattested word orders means we explicitly represent negative data.

We can define the problem in two slightly different ways, as a classification of types or a classification of tokens. In classifying language types, we try to assign each language type to a correct frequency value. Each type to be classified is unique, which yields 24 data points, for this universal. In developing a model based on a subset of the data, we are guaranteed that the new test data will be completely unseen.

In classifying tokens, we construct an experimental situation which corresponds to the real sampling. Each language type is represented by a variable number of languages. Some of the types are represented by many languages (those that are frequent), in our representation many instances of a given feature vector, other types will be represented by fewer languages. These differential frequencies are represented in the training by repeating each example the number of times indicated in Dryer's frequency counts by genera. So, for example, the vector that represents the word order N Dem Adj Num, attested in four genera, is repeated four times. Unattested word orders will be explicitly represented as negative data. (That is, unattested word orders are explicitly represented by one training exemplar.)[4] This set up has many more data points (214 in total) and it could happen that the test set contains examples of word orders that have also been seen at training time.

Figure 5 summarises the experimental setup. The three predictive regimes, ten-fold cross-validation, and the learning methods will be explained in the next section.

4.2 *Models*

Once the data are encoded in an appropriate way, we need to reproduce Cinque's way of assigning markedness values (fitting the weights), done by hand, or Cysouw's way of fitting the model to the

[4] This is a representational choice that allows us to represent negative data, as is common in supervised learning. Conceptually, this amounts to giving unattested word orders a (negative) observation in the training set. This means that we consider unattested data as data that we have not yet seen and that belong to a qualitatively different frequency class from rare data.

- **Type-based encoding:** each language type as positive or negative piece of data, possible or impossible word order.
- **Token-based encoding:** token-based classification encodes frequency of languages (notion of markedness), following Dryer's frequency counts based on genera, as size of sample in the training set.
- **Ten-fold cross-validation**
- **Three predictive regimes:**
  - two-way: possible, impossible;
  - four-way: very frequent, frequent, rare, unattested;
  - seven-way: two levels of very frequent, two levels of frequent, two levels of rare; one for unattested.

Figure 5: Summary of materials and method.

data. Cinque's and Cysouw's methods consist, manually or automatically, in assigning weights to reproduce the observed frequencies of possible and impossible values, with as close a fit as possible.

We will then test the predictive ability of these weighted explanations on data not seen at training time. In this set up, formally, we say that a computer program learns from experience E with respect to some task T and performance measure P, if its performance at task T, as measured by P, improves with E. In our case, the training experience E will be provided by a database of correctly classified language types or tokens; the task T consists in classifying word order types or tokens unseen in E into predetermined frequency classes; and the performance measure P will be defined as the percentage of types or tokens correctly classified. This learning paradigm is called supervised learning, because of the training phase, in which the algorithm is provided with examples and the correct answers. In the testing phase, these rules or probabilities are applied to additional data, not included in the training phase. The accuracy of classification on the test set indicates whether the rules or probabilities developed in the training phase are general enough, yielding good test accuracy, or are too specific to the training set to generalise well to other data.

There are numerous algorithms for learning the weights of a model in a supervised setting, and many regimes for training and testing such algorithms. In the following experiments, we use two probabilistic learning algorithms – Naive Bayes and the Weighted Average One-dependence Estimator – and n-fold cross-validation as the

Figure 6: Naive Bayes classifier.

Assume target function $f : X \rightarrow V$, where each instance x is described by attributes $\langle a_1, a_2 \dots a_n \rangle$.
Most probable value of $f(x)$ is:

$$v = \operatorname*{arg\,max}_{v_j \in V} P(v_j | a_1, a_2 \dots a_n), \quad (1)$$

$$v = \operatorname*{arg\,max}_{v_j \in V} \frac{P(a_1, a_2 \dots a_n | v_j) P(v_j)}{P(a_1, a_2 \dots a_n)} \quad (2)$$

$$= \operatorname*{arg\,max}_{v_j \in V} P(a_1, a_2 \dots a_n | v_j) P(v_j). \quad (3)$$

Naive Bayes assumption:

$$P(a_1, a_2 \dots a_n | v_j) = \prod_i P(a_i | v_j) \quad (4)$$

Naive Bayes classifier: $v_{NB} = \operatorname*{arg\,max}_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$

training and testing protocol (Russel and Norvig 1995; Webb *et al.* 2005).

The Naive Bayes algorithm is based on Bayes theorem and is defined in Figure 6. In this method, the objective of training is to learn the most probable word order type given the probability of each vector of features (see equation (1) in Figure 6). This probability is decomposed, according to Bayes rule, into the probability of the features given the word order and the prior probability of the word order itself (see equations (2) and (3) in Figure 6).

This method is chosen because it is a simple generative probabilistic model. Its generative probabilistic aspect provides a mathematically well-founded framework to predict frequencies and combine attributes. In a generative probabilistic setting, the typological frequencies are the expression of an underlying generative probabilistic model – the probabilistic independent variables – that give rise to the observed dependent variable – the frequency. The simplicity of the model has two justifications: on the one hand, the simplest models provide the strongest theories, by Occam's razor; on the other hand, a simple model allows a clear interpretation of the outputs and of the results.

In a classification task, we want to predict the class, in our case the frequency of the word order (for example, very frequent, frequent, rare, none), based on some descriptively pertinent features of the problem. The most noticeable feature of Naive Bayes is the very strong conditional independence assumption across features (see equation (4) in Figure 6). In our case, this assumption represents the intuition that the principles used to build word orders are independently motivated, and therefore they should be able to combine freely. This is a strong assumption that has important theoretical consequences. To verify its validity, then, we also experimented with a more complex model where properties are not assumed to be independent of each other. The model, called an averaged weighted one-dependence estimator (WAODE), assumes dependence from only one attribute at a time, taking the weighted average of the results of all the attributes.

To avoid excessive dependence of the results on a specific partition of the data, we use cross-validation. Cross-validation is a training and testing protocol in which the data is randomly partitioned into n parts, and then the learner is run n times, using $n-1$ partitions for training and the remaining one for testing. At each run of the learner, a different partition is chosen for testing. The performance measure is averaged over all n experiments.

Finally, the results will be compared to an uninformed baseline which consists in assuming that all word orders belong to the most frequent class. The baseline is helpful in indicating whether the models learn anything beyond mere frequency effects.

5 RESULTS AND DISCUSSION

We are now in a position to run the experiment. We run a 10-fold cross-validation, using a Naive Bayes classifier. We use the widely-used, open-source Weka data mining software.[5] Table 3 shows the results of the experiment, as the proportion of correct answers (percent accuracy).[6] In comparing these numbers, the discussion in the introduction should be borne in mind which indicated that models

[5] `http://weka.wikispaces.com/`

[6] As usual, accuracy is defined as the number of correctly classified items over the total number of items.

Table 3:
Percent (rounded accuracy) of languages or language types classified in the right frequency class. Italics indicate lower than baseline results.

	Naive Bayes					
	Type (24)			Token (214)		
	Two	Four	Seven	Two	Four	Seven
Cinque	88	58	42	97	87	89
Cysouw	*67*	*21*	66	*93*	90	68
Dryer	92	54	63	97	93	71
Baseline	71	50	38	97	47	28

with more parameters have more degrees of freedom and can fit the data better, but at the cost of greater complexity. At comparable performance levels, then, smaller models are usually preferred. By the same reasoning, small models that achieve lower performance than their competitors can often improve results by adding factors. As can be seen by the accuracy results, the models' generalisation is far from perfect, at the level of language types (shown in the left panel). In the binary classification, possible or impossible languages, almost 10% of the data are incorrectly classified. See for example the results on two-way type-based classification of both Cinque and Dryer. Some of the models of type-based classification have performances below or equal to the baseline: the model does not learn. This result illustrates the lesson that models need to be tested on external data; conclusions based on the data used to develop the models are often overly optimistic. Token-based classification yields better results, especially in the four-way classification, with a small number of factors.

5.1 *Analysis of results*

We concentrate now on a more detailed analysis of the models, starting with Cinque's model. The aggregated accuracy results shown in Table 3 can be disaggregated into more informative subcases, by looking at precision and recall by frequency type and by looking at confusion matrices.[7] All the mistakes, as indicated by the results per class and by the confusion matrix, shown in Tables 4 and 5, fall in the *frequent, rare*, and *none* category. Interestingly, most mistakes tend to

[7] As usual, we use the measures of precision and recall. Precision is the number of correctly classified items over the total number of items proposed by the algorithm as belonging to a given class; recall is the number of correctly classified items over the total number of items that should have been found in a given class; and F is their harmonic mean. Confusion matrices indicate the correct output by rows and the model's predictions by columns.

	Naive Bayes Results		
	Precision	Recall	F
Very Frequent	91	100	95
Frequent	85	86	85
Rare	91	57	70
None	56	71	62

Table 4:
Percent precision, recall and F measure by frequency class of token-based Naive Bayes classifier for Cinque's model.

	Confusion Matrix			
	Very Frequent	Frequent	Rare	None
Very Frequent	101	0	0	0
Frequent	10	61	0	0
Rare	0	11	20	4
None	0	0	2	5

Table 5:
Confusion Matrix of token-based Naive Bayes classifier for Cinque's model.

classify the tokens in a class of higher frequency than the correct one; only four of the rare cases are mistakenly classified as unattested. This shows that the attributes associated with frequent events dominate the classification.

The Naive Bayes confusion matrix by frequency class indicates that very frequent orders and unattested word orders are overestimated (Recall > Precision), while frequent and rare word orders are underestimated (Precision > Recall). The fact that the F-measure decreases with the frequency of the class indicates that the model is not a good predictor of cases that are rarely attested in the training data.

Even more informative are the actual probabilities learnt by the model. If we look at the joint probability distribution of the attributes and their values, shown in Table 6, we can see that there is a very strong association among one value of the three merge attributes (first, second, and third) and one class of frequency: *first:AN*, *second:NumA*, and *third:DNum* are indicators of the difference between all three attested frequency classes and the unattested one. The attributes *complete* and *partial* are not as informative about the frequency distinctions.

We can also calculate the probabilities of different aspects of the model by marginalising out some of the details of the distribution. If we marginalise out the values by frequency, we find that partial and complete movement have very different distributions, as shown

Table 6: Cinque's joint probability Naive Bayes tables.

		Very Frequent	Frequent	Rare	None
First	AN	0.96	0.95	0.85	0.25
	DN	0.01	0.013	0.025	0.25
	ND	0.01	0.013	0.025	0.17
	NNum	0.01	0.013	0.025	0.17
	NumN	0.01	0.013	0.075	0.17
Second	AD	0.01	0.012	0.24	0.15
	DA	0.01	0.012	0.097	0.23
	DNum	0.01	0.012	0.073	0.23
	NA	0.95	0.93	0.76	0.08
	NumD	0.01	0.012	0.24	0.15
	NumN	0.01	0.012	0.24	0.15
Third	AD	0.01	0.012	0.24	0.23
	AN	0.01	0.012	0.073	0.08
	ANum	0.01	0.012	0.24	0.23
	DNum	0.95	0.93	0.76	0.08
	NumA	0.01	0.012	0.24	0.15
	NumD	0.01	0.012	0.097	0.23
Partial	not	0.43	0.013	0.28	0.64
	np	0.009	0.29	0.38	0.09
	of-who-pp	0.009	0.013	0.20	0.09
	whose-pp	0.55	0.68	0.13	0.18
Complete	not	0.43	0.53	0.46	0.73
	np	0.009	0.16	0.15	0.09
	of-who-pp	0.009	0.29	0.15	0.09
	whose-pp	0.55	0.013	0.23	0.09

Table 7: Probability distributions of feature values by type of movement.

	not	np	of-who-pp	whose-pp
Partial	0.34	0.19	0.08	0.39
Complete	0.54	0.10	0.13	0.22

in Table 7.[8] If we sum up the probabilities and compare all types of movement operations (the last three columns) to no movement, we find that the partial movement operation is twice as probable as no movement, while complete movement is a little less probable than no complete movement. This shows that while no movement is preferred to complete movement, as predicted by Cinque's theory, partial movement is more probable than no partial movement, and also more probable than complete movement. These two results

[8] Recall that movement of the *pictures of who* type is coded as *of-who-pp* and *whose picture* is coded as *whose-pp*.

	Very Frequent		Frequent		Rare		None	
	Y	N	Y	N	Y	N	Y	N
NA-adjacency	0.99	0.01	0.99	0.01	0.40	0.60	0.33	0.67
N-edge	0.99	0.01	0.16	0.84	0.49	0.51	0.55	0.45
Dem-edge	0.99	0.01	0.55	0.45	0.51	0.49	0.11	0.89

Table 8: Cysouw's joint probability Naive Bayes tables.

	Naive Bayes Results		
	Precision	Recall	F
Very Frequent	99	100	99
Frequent	83	100	91
Rare	81	60	69
None	0	0	0

Table 9: Percent precision, recall and F measure by frequency class of token-based Naive Bayes classifier for Cysouw's model.

are not expected, as complete movement is supposed to be easier than partial movement, so that one could expect it to occur more often.

We can also observe how partial and complete movement types pattern across frequency levels. There are different types of frequent word orders, and even more types of rare word orders. If we look at the distribution of types of movement for frequent and rare word orders, we see the patterns shown in the two central columns (Frequent, Rare) of the last two sets of rows (Partial, Complete) in Table 6. Partial movement is not always more frequent and complete movement is not always less frequent. The noticeable differences in distributions indicate that all these distinctions (partial, complete) and their four levels are needed for accurate classification.

The same analysis of results can be applied to Cysouw's model. In Table 8, we can see that *NA-adjacency* distinguishes very frequent and frequent word orders from rare and unattested word orders, but does not distinguish within these two groups; *N-edge* distinguishes all four classes; *Dem-edge* makes a three-way distinction, it distinguishes very frequent, from frequent and rare, from unattested. The most prominent results shown in the disaggregated precision and recall measures by class concerns unattested word orders, as indicated in Table 9. Cysouw's model does not appear to be able to predict this frequency class. The confusion matrix, shown in Table 10, indicates that unattested word orders are confused with rare word orders, but also with frequent word orders. Rare word orders also show several errors, confused with frequent and, in two cases, with very frequent word orders.

Table 10: Confusion Matrix of Naive Bayes classifier for Cysouw's model.

	Very frequent	Frequent	Rare	None
Very frequent	101	0	0	0
Frequent	0	71	0	0
Rare	2	12	21	0
None	0	2	5	0

Table 11: Dryer's joint probability Naive Bayes tables.

	Very Frequent		Frequent		Rare		None	
	Y	N	Y	N	Y	N	Y	N
Symmetry1	0.99	0.01	0.84	0.16	0.62	0.38	0.22	0.78
Symmetry2	0.99	0.01	0.99	0.01	0.54	0.46	0.55	0.45
Asymmetry	0.99	0.01	0.99	0.01	0.92	0.08	0.11	0.89
U18	0.99	0.01	0.99	0.01	0.65	0.35	0.67	0.33
Harmony	0.99	0.01	0.16	0.84	0.49	0.51	0.55	0.45

This model makes fewer mistakes, but appears to have a higher degree of confusion across frequency types than Cinque's model.

The analysis of Dryer's model shows different patterns of distributions and errors from the other two models. If we look at the joint probability distributions associated with the attributes in Dryer's model, shown in Table 11, we can observe that the principle *Symmetry1* discriminates all frequency classes, while neither the principles *Symmetry2, Asymmetry* nor *U18* make a clear distinction between very frequent and frequent word orders, and between rare and unattested word orders. The *Harmony* principle, on the other hand, does discriminate among all frequency classes, often in the opposite direction from the principle *Symmetry1*. The most surprising observation that emerges from these probabilites is that frequent word orders are observed to be frequent, despite the fact that they are disharmonic ($P = 0.17$ for the probability of exhibiting the *Harmony* property for frequent word orders, compared to $P = 0.87$ for those not exhibiting this property). Table 12 shows that this model is affected by frequency effects, as shown by the fact that frequent word orders are overestimated (Precision > Recall), while rare word orders are underestimated (Precision < Recall). Table 13 shows that there are twice as many errors confusing more frequent with less frequent word orders than the reverse (11 vs. 5). The table also shows that frequent and rare orders are confused, and that rare and unattested orders are also confused.

These analyses of the errors show that, once tested in a precise learner, the attributes that define these three theories do not always

	Naive Bayes Results		
	Precision	Recall	F
Very Frequent	100	100	100
Frequent	94	84	90
Rare	73	86	79
None	86	86	86

Table 12:
Percent precision, recall, and F measure by frequency class of token-based Naive Bayes classifier for Dryer's model.

	Very frequent	Frequent	Rare	None
Very frequent	101	0	0	0
Frequent	0	61	10	0
Rare	0	4	30	1
None	0	0	1	6

Table 13:
Confusion Matrix of Naive Bayes classifier for Dryer's model.

behave as expected. For example, in Cinque's model, complete movement is less likely than partial movement, while in Dryer's model some of the attributes do not discriminate the typological frequency classes.[9] Also, all the models make mistakes when used predictively. Because the Naive Bayes model is predicated on a strong independence assumption of the attributes, we turn to verifying if this assumption is valid for our data.

5.2 *Validating the independence assumption*

As a control of the independence assumption in the Naive Bayes model, we also learn the data with a probabilistic classifier that relaxes the strong independence assumption. The model, called an averaged weighted one-dependence estimator (WAODE), assumes dependence from only one attribute at a time, taking the weighted average of all the possible dependencies. What is relevant here is that this constitutes a minimally different model from a Naive Bayes classifier, so that only the assumption of independence of attributes is changed across the two models. For Cinque's and Dryer's models, results are much better, as shown in Table 14. In particular, the classifiers no longer mistake systematically the frequent word orders, as shown in Tables 15, 16, and 17, reporting the confusion matrices. However, here again the accuracy, while very high, is not perfect. This demonstrates that a true separate test set is needed to assess the real generality of the

[9] I thank one of the reviewers for pointing out, correctly, that this result actually means that Dryer's model could have fewer attributes, hence could be made more economical, without loss in predictive power.

Table 14: Percent of languages classified in the right frequency class, for a token-based four-way classification.

	WAODE classifier				Naive Bayes
	Precision	Recall	F	Acc	Acc
Cinque	94	93	93	93	87
Cysouw	87	90	88	90	90
Dryer	96	96	96	96	93

Table 15: Confusion Matrix of WAODE classifier for Cinque's model.

	Very frequent	Frequent	Rare	None
Very frequent	101	0	0	0
Frequent	0	71	0	0
Rare	0	10	23	1
None	0	0	2	5

Table 16: Confusion Matrix of WAODE classifier for Dryer's model.

	Very frequent	Frequent	Rare	None
Very frequent	101	0	0	0
Frequent	0	71	0	0
Rare	0	7	28	0
None	0	0	1	6

Table 17: Confusion Matrix of WAODE classifier for Cysouw's model.

	Very frequent	Frequent	Rare	None
Very frequent	101	0	0	0
Frequent	0	71	0	0
Rare	1	12	21	0
None	0	2	5	0

proposed models. Cysow's model, on the other hand, has the same accuracy (and same confusion matrix) in the two models, which shows that the parameters in this model are indeed independent.

The fact that a classifier that makes weaker independence assumptions about its attributes yields better performance than Naive Bayes, which assumes conditional independence of the attributes, indicates that the attributes are not independent. These attributes are supposed to be the primitive, independently motivated – in a different sense of the word *independent* – operations and properties of the different linguistic proposals that give rise to the different word orders. Finding a statistical dependence among them indicates that part of the explanation of the data is given by the interaction of the factors, interaction that cannot be independently motivated, as it is specific to these data. This means that part of the explanation provided by the linguistic models rests on interactions other than those operations that can be justified on general theoretical grounds.

6 CONCLUSIONS

This paper has shown in detail how simple computational learning paradigms can help test and compare theories about universals. The process of finding probabilities automates and makes mathematically precise the assignment of weights that we find in proposals about language universals, but does not change the logic of these proposals. The added value of this procedure is two-fold. On the one hand, we use a mathematically well-defined probabilistic framework, so that combination of factors, ranking, and optimisation processes are well-defined. On the other hand, the evaluation rests on the use of unseen data, so that the quantitative results are a measure of generalisation. This method, then, constitutes a well-defined procedure to estimate the weights of the operations and aspects of the models and to compare their generalisation capabilities, with sometimes interesting results. For example, we uncover the fact that the properties of the models are interdependent, and hence not theoretically fully independently motivated. Future work lies in developing more accurate models for more complex or more comprehensive problems.

7 REFERENCES

Steven ABNEY (2011), Data-intensive experimental linguistics, *Linguistic Issues in Language Technology (LILT)*, 6(2):1–27.

Guglielmo CINQUE (2005), Deriving Greenberg’s Universal 20 and its exceptions, *Linguistic Inquiry*, 36(3):315–332.

Jennifer CULBERTSON and Paul SMOLENSKY (2012), A Bayesian model of biases in artificial language learning: The case of a word-order universal, *Cognitive Science*, 36(8):1468-1498.

Jennifer CULBERTSON, Paul SMOLENSKY, and Geraldine LEGENDRE (2012), Learning biases predict a word order universal, *Cognition*, 122(3):306–329.

Michael CYSOUW (2010a), Dealing with diversity: towards an explanation of NP word order frequencies, *Linguistic Typology*, 14(2):253–287.

Michael CYSOUW (2010b), On the probability distribution of typological frequencies, in *Proceedings of the 10th and 11th Biennial conference on the* *mathematics of language*, MOL'07/09, pp. 29–35, Springer-Verlag, Berlin, Heidelberg.

Matthew DRYER (2006), The order demonstrative, numeral, adjective and noun: an alternative to Cinque, `http://attach.matita.net/caterinamauri/sitovecchio/1898313034_cinqueH09.pdf`. Accessed on 19th August, 2015.

Matthew DRYER (2005), "Genealogical language list", in Haspelmath, Martin and Dryer, Matthew and Gil, David and Comrie, Bernard (eds.), *The World Atlas of Language Structures*, Oxford University Press, Oxford, 584-644.

Matthew S. DRYER (1992), The Greenbergian word order correlations, *Language*, 68:81–138, doi:10.2307/416370.

Michael DUNN, Simon J. GREENHILL, Stephen C. LEVINSON, and Russell D. GRAY (2011), Evolved structure of language shows lineage-specific trends in word-order universals, *Nature*, 473:79–82.

Richard FUTRELL, Kyle MAHOWALD, and Edward GIBSON Large-scale evidence of dependency length minimization in 37 languages, *Proceedings of the National Academy of Sciences of the United States of America*, 112(33):10336-10341, doi:10.1073/pnas.1502134112.

Joseph H. GREENBERG (1966), *Language universals*, Mouton, The Hague, Paris.

Richard KAYNE (1994), *The antisymmetry of syntax*, MIT Press, Cambridge, MA.

Stuart RUSSEL and Peter NORVIG (1995), *Artificial intelligence: a modern approach*, Prentice Hall Series in Artificial Intelligence, Prentice Hall, Upper Saddle River, NJ.

Mark STEEDMAN (2011), Greenberg's 20th: the view from the long tail, unpublished manuscript, University of Edinburgh.

G. I. WEBB, J. BOUGHTON, and Z. WANG (2005), Not so Naive Bayes: aggregating one-dependence estimators, *Machine Learning*, 58(1):5–24.

How to keep the HG weights non-negative: the truncated Perceptron reweighing rule[*]

Giorgio Magri
SFL (CNRS and University of Paris 8) and UiL-OTS (Utrecht University)

ABSTRACT

The literature on error-driven learning in Harmonic Grammar (HG) has adopted the *Perceptron* reweighing rule. Yet, this rule is not suited to HG, as it fails at ensuring non-negative weights. A variant is thus considered which truncates the updates at zero, keeping the weights non-negative. Convergence guarantees and error bounds for the original Perceptron are shown to extend to its truncated variant.

Keywords: Harmonic Grammar, error-driven learning, Perceptron, convergence

1 INTRODUCTION

Language learning is the process of selecting a grammar from a given typology of grammars based on some linguistic data. Assume that the learner maintains a current grammar representing its current hypothesis on the target adult grammar it is being trained on. Training data come in a stream. Whenever the current grammar makes an error on the current piece of training data, it is updated to a slightly different one. The current piece of data is then discarded and the learner waits for the next piece of training data to evaluate the performance of the updated grammar. This learning scheme is called *error-driven* because the learning dynamics is driven by the errors made on the incoming stream of data. This scheme has been thoroughly investigated in the machine learning literature (where it is commonly called *online learning*; for a review, see Kivinen 2003; Cesa-Bianchi and Lugosi 2006,

[*] I would like to thank Paul Boersma and Nazarré Merchant for useful comments. The research reported in this paper has been supported by a Marie Curie Intra European Fellowship (Grant agreement number: PIEF-GA-2011-301938).

chapters 11, 12; and Mohri *et al.* 2012, ch. 7). Within the language acquisition literature, this learning scheme has been endorsed at least since Wexler and Culicover (1980) for two reasons. First, an error-driven learner describes a sequence of grammars in typological space and thus provides a tool to model child acquisition paths. Second, an error-driven learner does not keep track of previously seen data (the current piece of data is discarded at the end of each iteration) and can thus be used to model the early stages of language acquisition prior to the development of the native language lexicon (such as the early acquisition of phonotactics; Hayes 2004).

The most basic question of the computational theory of error-driven learning concerns *convergence*: is it possible to guarantee that the learner only makes a finite number of errors, so that it describes a finite sequence of grammars in typological space? Convergence is crucial because it means that the learner eventually settles on a final grammar which will never be further updated and thus counts as the grammar *learned* by the algorithm. This paper focuses on convergence of error-driven learning within the framework of *Harmonic Grammar* (HG; Legendre, Miyata, and Smolensky 1998b,a; Smolensky and Legendre 2006).

Within the HG framework, the typology of grammars is parameterized through an assignment of *weights* to a given, finite set of *constraints* which extract the relevant properties of the linguistic data. Whenever the HG error-driven learner makes an error, the constraints are slightly reweighed. The recent HG computational literature has adopted the *Perceptron* (or *delta*) reweighing rule (Pater 2008; Jesney and Tessier 2011; Coetzee and Pater 2008, 2011; Coetzee and Kawahara 2013; Boersma and Pater to appear, among many others). According to this rule, a certain amount is added to certain weights and subtracted from others. This reweighing rule comes with convergence guarantees, reviewed in Section 2: the number of errors is always finite and can be bounded in terms of certain "geometric" properties of the training data (Rosenblatt 1958, 1962; Block 1962; Novikoff 1962; Minsky and Papert 1969; Cesa-Bianchi and Lugosi 2006, chapters 11, 12; Mohri *et al.* 2012, ch. 7).

Despite current practice, the Perceptron is *not* suited to HG. Crucially, HG requires the weights to be non-negative, in order to avoid undesired typological predictions. Yet, the Perceptron does not enforce non-negativity of the final weights, since the current weights are decreased (as well as increased) throughout learning. A simple solution to this problem is a variant of the Perceptron reweighing rule which truncates the updates at zero, thus ensuring non-negativity of the final weights (Boersma and Pater to appear; Boersma and van Leussen 2014). Although a run of the original and the truncated Perceptron can differ substantially, Section 3 shows that a run of the truncated Perceptron can be "mimicked" with a run of the original Perceptron on a slightly different sequence of data. Convergence guarantees thus extend from the original to the truncated Perceptron. This observation yields the first convergence guarantee for an HG error-driven learner consistent with HG's restriction to non-negative weights. This result is constraint-independent, namely it follows from the HG mode of constraint interaction and thus holds for any constraint set. Section 4 extends the reasoning to the stochastic implementation of HG error-driven learning and to the noisy learning setting.

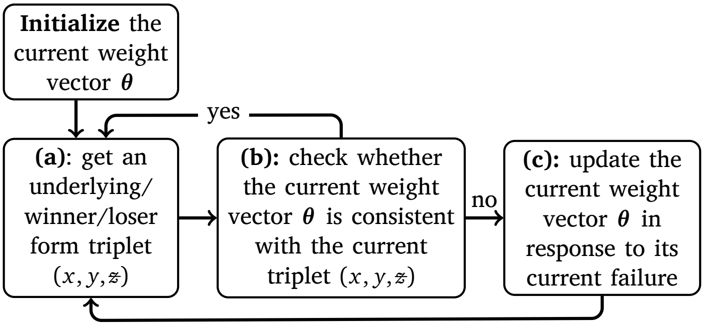
2 THE *ORIGINAL* PERCEPTRON HG LEARNER

This Section reviews the implementation of error-driven learning used in the current HG literature, based on the *Perceptron* reweighing rule.

2.1 *Algorithmic core*

Within HG, the typology of grammars is parameterized by an assignment of *weights* $\theta_1, \ldots, \theta_k, \ldots, \theta_n$ to a given collection of n phonological constraints $C_1, \ldots, C_k, \ldots, C_n$. These weights are collected together into a *weight vector* $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$. The error-driven learning scheme can then be made explicit in HG as in (1).

(1)


Initialize the current weight vector θ
yes
(a): get an underlying/ winner/loser form triplet (x, y, ƶ)
(b): check whether the current weight vector θ is consistent with the current triplet (x, y, ƶ)
no
(c): update the current weight vector θ in response to its current failure

The algorithm maintains a current vector $\boldsymbol{\theta}$ of constraint weights. The analyses reported in this paper are independent of how these weights are initialized; for concreteness, I assume throughout that they are all initialized to zero (the extension to arbitrary initial weights is straightforward). The current weights are then updated by looping through the three steps (1a)–(1c), described in detail below.

2.2 *Data provided at step (1a)*

At step (1a), the learner receives a piece of data. At a minimum, this piece of data consists of a surface form y, say some form which is licit according to the phonotactics corresponding to the target grammar the learner is being trained on. In some applications, the corresponding underlying form x can be assumed to be provided as well. In other applications, the learner needs to be endowed with an additional subroutine to reconstruct the underlying form (e.g., set the current underlying form x identical to the current surface form y; Prince and Tesar 2004; Hayes 2004; Magri 2015). Yet, the analyses reported in this paper are independent of the subroutine for the choice of the underlying form, which I thus assume to be provided in some arbitrary way along with the surface form at step (1a). The mapping (x, y) of the underlying form x to the surface form y must beat the mapping $(x, \cancel{z})$ of x to any other loser candidate $\cancel{z}$ (loser candidates are stricken out as a mnemonic) according to the target grammar the learner is being trained on. The learner needs to focus on one such loser candidate $\cancel{z}$. Usually, this loser candidate is chosen through a proper subroutine.[1] Yet, the analyses reported in this paper are independent of the subroutine for the choice of the loser form, which I thus assume to be provided in some arbitrary way at step (1a) as well. In the end, I assume that the learner is fed at step (1a) a piece of data which consists of an underlying/winner/loser form triplet $(x, y, \cancel{z})$. The collection of these triplets is called the *training set* (each triplet from the training set can of course be fed multiple times to the learner).

[1] A reasonable choice is to set the current loser $\cancel{z}$ equal to the candidate which is predicted to win according to the grammar corresponding to the current weight vector $\boldsymbol{\theta}$. With this choice of the current loser, the following step (1b) can be reformulated as follows: "check whether the intended winner y coincides with the predicted winner $\cancel{z}$".

2.3 *Consistency condition checked at step (1b)*

At step (1b), the learner checks whether the current weight vector $\boldsymbol{\theta}$ is *consistent* with the current underlying/winner/loser form triplet $(x, y, \mathfrak{z})$, namely whether the HG grammar corresponding to the current weight vector $\boldsymbol{\theta}$ manages to make the intended winner y beat the intended loser $\mathfrak{z}$ for the underlying form x. To start, assume that all constraints are *binary*, namely they assign at most one violation. In this case, the condition that the intended winner y beats the intended loser $\mathfrak{z}$ boils down to the condition (2). This condition says that the sum of the current weights $\theta_1, \theta_2, \ldots$ of the *winner-preferring* constraints collected into the set W (namely those constraints which assign fewer violations to the winner y than to the loser $\mathfrak{z}$) is larger than the sum of the weights of the *loser-preferring* constraints collected into the set L (namely those constraints which assign fewer violations to the loser $\mathfrak{z}$ than to the winner y).

(2) $$\sum_{h \in W} \theta_h > \sum_{k \in L} \theta_k$$

In the general case of arbitrary (possibly non-binary) constraints, the consistency condition generalizes to (3). The *violation difference* of constraint C_k is the difference between the number of violations $C_k(x, \mathfrak{z})$ it assigns to the loser $\mathfrak{z}$ minus the number of violations $C_k(x, y)$ it assigns to the winner y (see appendix A.1 for discussion). Condition (3) thus requires the average of the constraint violation differences weighted by the current weights θ_k to be strictly positive.

(3) $$\sum_{k=1}^{n} \Big(\underbrace{C_k(x, \mathfrak{z}) - C_k(x, y)}_{\substack{\text{violation difference} \\ \text{of constraint } C_k}} \Big) \theta_k > 0$$

In the case of binary constraints, the consistency condition (3) indeed reduces to (2).

2.4 *Update performed at step (1c)*

If the consistency condition (2)/(3) is satisfied, the current weight vector already predicts that the current winner y beats the current loser $\mathfrak{z}$. The learner thus has nothing to learn from the current comparison and loops back to step (1a). Otherwise, the current weight vector $\boldsymbol{\theta}$ needs to be updated at step (1c) in response to its current failure. To start, assume that the constraints are all binary. Failure of condition (2) suggests that the weights corresponding to the winner-preferring (loser-preferring) constraints are too small (too large). One reasonable update strategy is thus (4), which promotes (demotes) the winner-preferring (the loser-preferring) constraints by a small amount, say 1 for concreteness; for an illustration, see (10) below.

(4) a. Increase the current weight of each winner-preferring constraint by 1;

b. decrease the current weight of each loser-preferring constraint by 1.

In the general case of arbitrary (possibly non-binary) constraints, the update rule (4) is generalized as in (5) (Jesney and Tessier 2011; Coetzee and Pater 2008, 2011; Coetzee and Kawahara 2013; Boersma and Pater to appear, among many others). If a constraint C_k is winner-preferring (loser-preferring), its violation difference $C_k(x,\text{ƶ})-C_k(x,y)$ is positive (negative) and its weight is therefore increased (decreased) by the update rule (5). In the case of binary constraints, the update rule (5) indeed reduces to (4).

(5) Update each current weight θ_k by adding the corresponding violation difference $C_k(x,\text{ƶ})-C_k(x,y)$.

After the update, the learner loops back to step (1a), waits for another piece of data, and starts all over again.

2.5 *Convergence*

Boersma and Pater (to appear) note that the HG reweighing rule (5) can be interpreted as the *Perceptron* (or *delta*) update rule. They thus reinterpret the convergence guarantees for the Perceptron (Rosenblatt 1958, 1962; Block 1962; Novikoff 1962; Minsky and Papert 1969; Cristianini and Shawe-Taylor 2000, Theorem 2.3; Cesa-Bianchi and Lugosi 2006, ch. 12; Mohri *et al.* 2012, ch. 7) as the following Theorem 1 on convergence of the HG error-driven learner.[2] See Magri (to appear) for discussion of the error bound (6).

[2] Boersma and Pater (to appear) call the learner just described the (deterministic) *HG-GLA*. I prefer the name *HG (Perceptron) error-driven learner*, thus keeping the acronym "GLA" for a specific implementation of OT error-driven learning, characterized by the fact that the promotion amount is set equal to the demotion amount.

Theorem 1 *The HG error-driven learner (1) with the HG update condition (3) and the Perceptron reweighing rule (5) converges: the number of errors is bounded by*

$$(6)\qquad \left(\frac{\textit{radius of the training data}}{\textit{margin of the training data}}\right)^2$$

when the training set consists of underlying/winner/loser form triplets which are all consistent with some HG grammar and have bounded violation differences.

Theorem 1 provides an error bound (6) which depends on some "geometric" properties of the training data, namely their *radius* and their *margin (of separability)*. Here is the idea in a nutshell. The training set consists of underlying/winner/loser form triplets $(x, y, \not{z})$. For each of these training triplets, collect into a vector the n violation differences assigned by the phonological constraints $C_1, \ldots, C_n$ to that triplet. The resulting vector can be thought of as a point in the cartesian n-dimensional space. The *radius* of the training data which appears in the numerator of (6) is the radius of the smallest sphere which contains all the vectors of constraint violation differences that the learner is trained on, as explained in Appendix A.2. Of course, the error bound (6) only makes sense provided that the radius in the numerator is finite, or equivalently that the violation differences are bounded, as indeed required by the statement of the Theorem. That is in particular the case if the number of violations assigned by the constraints is upper bounded by some constant.

The precise definition of the *margin (of separability)* which appears in the denominator of (6) is somewhat involved and is therefore relegated to Appendix A.3. The following intuitive illustration suffices for the rest of the paper. Theorem 1 assumes that the training set is consistent with some HG grammar. Yet, consistent training sets can differ in their *degree of consistency*. The training set has a *high* degree of consistency if it is consistent with a certain HG grammar and remains consistent when the corresponding weight vector is tampered with. The training set has instead a *small* degree of consistency if even a slight modification of any consistent weight vector affects consistency. The margin which appears in the denominator of (6) can be interpreted as the degree of consistency of the training set. Intuitively, a training set with a large degree of consistency should be easy to learn:

it should be easy to shoot at a consistent weight vector. A training set with a small degree of consistency should instead be hard to learn: a careful aim is required to shoot precisely at a consistent weight vector. The error bound (6) formalizes this intuition: a training set with a high (low) degree of consistency has a large (small) margin, yielding a small (large) error bound (6) which provides guarantees for better (worse) performance of the HG learner.

3 THE *TRUNCATED* PERCEPTRON HG LEARNER

This section explains why the problem of convergence for HG error-driven learning is still open in the literature and provides a simple and principled solution.

3.1 *The problem of non-negative weights*

Constraints in HG are always interpreted as expressing penalties, never rewards. Hence, constraint weights need to be enforced to satisfy the non-negativity condition (7) in order for HG to avoid undesired typological predictions, whereby less marked structures are mapped to more marked ones (Legendre *et al.* 2006; Keller 2000).

(7) $\theta_1, \dots, \theta_n \geq 0$

Here is an elementary counterexample which illustrates the importance of this non-negativity condition. Suppose that the constraint set contains the markedness constraint NoVoice against voiced obstruents and the identity faithfulness constraint Ident[voice] for voicing. Suppose furthermore that the underlying form /ta/ comes with the two surface candidates [ta] and [da]. If the two constraints are allowed to take on negative weights (say $\theta_{\text{NoVoice}} = -3$ and $\theta_{\text{Ident}} = -1$), the corresponding HG grammar maps the voiceless stop to a voiced one, whereby an unfaithful mapping yields no gain in markedness.

Despite the non-negativity condition (7) being a crucial component of HG's mode of constraint interaction, the Perceptron update rule (4)/(5) used in the current literature does not in any way guarantee that the current and final weights entertained by the algorithm satisfy this non-negativity condition (7). Even if the current weights are initialized to large initial values, there is no guarantee that they will never drop below zero, as the number of updates – and thus in particular the number of demotions (4b) – crucially depends on the size of the initial weights. Furthermore, certain modeling applications have been argued to require certain constraints to start with initial weights equal to zero, namely to start right at the edge of the forbidden zone. For instance, Jesney and Tessier (2011) argue that input-output faithfulness constraints need to start with null initial weights, in order to prevent gang-up effects that might foul the learner into learning phonotactically unrestrictive final weights. In conclusion, the Perceptron reweighing rule (4)/(5) does not yield a proper HG error-driven learner. The rest of this section develops a solution to this problem.[3]

3.2 *The truncated Perceptron reweighing rule*

To start, assume for concreteness that all constraints are binary. A natural strategy to enforce non-negativity of the current and final weights is to switch from the *original* Perceptron update rule (4) to the *truncated* Perceptron update rule (8). The two update rules coincide as long as the current weights stay non-negative. But when the original update rule (4) would demote a certain weight below zero, the truncated rule (8) leaves that weight unchanged; for an illustration, see (12) below.

[3] A different solution to this problem is to use the *Winnow* algorithm instead of the Perceptron algorithm (Littlestone 1988). In fact, Winnow adopts a *multiplicative* update rule (rather than the Perceptron's *additive* update rule) and therefore effectively keeps the weights non-negative. Unfortunately, convergence guarantees for Winnow only hold when the amount of reweighing (also called the *plasticity* or the *step size*) has been properly chosen in a way that crucially depends on the margin of the training data. Since the margin is not known beforehand, the algorithm needs to be supplemented with a procedure to estimate the margin online, making the overall implementation more complex. Despite this difficulty, it might be worth exploring the use of Winnow's reweighing rule for HG error-driven learning. In fact, Boersma and Pater (to appear) report simulation results with a reweighing rule which is very similar to Winnow's (it only differs because the current weights are not normalized, contrary to what is prescribed by Winnow). Although the variant tested in Boersma and Pater's simulations has no guarantees of convergence (normalization of the weights plays a crucial role in Winnow's convergence proof), they report that the number of errors is significantly smaller than with the Perceptron on their test cases. Indeed, Winnow and the Perceptron have been compared extensively in the machine learning literature (Kivinen, Warmuth, and Auer 1997), with the two update rules outperforming each other on different types of training sets.

(8) a. Increase the current weight of each winner-preferring constraint by 1;

b. decrease the current weight of each loser-preferring constraint by 1, *unless that would make that weight negative, in which case do not modify that weight.*

In the general case of arbitrary (possibly non-binary) constraints, the truncated Perceptron reweighing rule becomes (9). This is the original update rule (5) apart from the additional "unless" clause in italics, meant to prevent the weights from ever turning negative.[4] In the case of binary constraints, (9) indeed reduces to (8).

(9) Update each current weight by adding the violation difference of the corresponding constraint, *unless that update would make that current weight negative, in which case do not modify that weight.*

Boersma and Pater (to appear, p. 19) and Boersma and van Leussen (2014, section 5) report encouraging simulation results with this truncated reweighing rule. But what about its theoretical guarantees? Theorem 1 guarantees that the learner with the *original* Perceptron update rule (4)/(5) can only make a finite number of errors and furthermore provides an explicit error bound. What about the truncated Perceptron update rule (8)/(9)? The two update rules are superficially similar; yet, they describe quite different algorithms. Indeed, suppose that the current weights are all initialized to zero. The original Perceptron update rule will then perform lots of demotions below zero that the truncated Perceptron is forbidden to mimic. As a result, the sequence of grammars entertained by the original Perceptron will turn out to be quite different from the sequence of grammars entertained by the truncated Perceptron. Is there any way to extend the theoretical guarantees that hold for the original Perceptron to its truncated variant?

3.3 *Sketch of the analysis on a concrete example*

Consider three constraints IDENTVOICEONSET (which requires preservation of voicing in onset position), IDENTVOICE (which requires

[4] A slight variant of (8)/(9) is as follows: when updating a weight would make that weight negative, instead of leaving that weight unchanged, set that weight equal to the smallest licit value, namely to zero. The analysis presented below trivially extends to this variant as well.

preservation of voicing in an arbitrary position), and NoVoice (which militates against obstruent voicing). Assume that the underlying form /da/ only comes with the two candidates [da] and [ta]. Suppose that at a certain iteration of the HG learner, the current weight of the constraint IdentVoiceOnset is equal to zero, thus barely satisfying the non-negativity condition (7). Suppose for concreteness that the current weights of the other two constraints IdentVoice and NoVoice are instead positive, say equal to 7 and 3 respectively, as indicated by the weight vector on the left hand side of (10).

(10) *Update by the original Perceptron:*

$$\begin{array}{r} \text{IdentVoiceOnset} \\ \text{IdentVoice} \\ \text{NoVoice} \end{array} \begin{bmatrix} 0 \\ 7 \\ 3 \end{bmatrix} \xrightarrow{(\text{/da/, [ta], } \cancel{\text{[da]}})} \begin{bmatrix} -1 \\ 6 \\ 4 \end{bmatrix}$$

Suppose that the learner is trained on a target grammar which bans voiced obstruents across the board. The learner is thus fed the underlying form /da/ together with the corresponding intended winner [ta] and the faithful loser ~~[da]~~ , as indicated by the label on top of the arrow in (10).

The markedness constraint NoVoice prefers the winner candidate [ta] while the two faithfulness constraints IdentVoiceOnset and IdentVoice prefer the loser candidate ~~[da]~~, as shown in (11).

(11)

Input: /da/	IdentVoiceOnset $\theta = 0$	IdentVoice $\theta = 7$	NoVoice $\theta = 3$
a. [ta]	∗(L)	∗(L)	
b. ☞ ~~[da]~~			∗(W)

The weight $\theta_{\text{NoVoice}} = 3$ of the winner-preferring constraint is not larger than the sum $\theta_{\text{IdentVoiceOnset}} + \theta_{\text{IdentVoice}} = 0 + 7$ of the weights of the two loser-preferring faithfulness constraints. Condition (2) therefore fails and the current weights need to be updated. The original Perceptron update rule (4) prescribes that the weights of the two loser-preferring constraints IdentVoiceOnset and IdentVoice each be decreased by 1 while the weight of the winner-preferring constraint NoVoice be increased by 1, obtaining the updated weight vector on the right hand side of (10). The weight of the loser-preferring constraint IdentVoiceOnset has thus dropped to the negative value −1, in violation of the non-negativity condition (7). If this were the final update, the learner would have effectively failed.

The update according to the truncated Perceptron update rule (9) in this same scenario is described in (12). The weight of the winner-preferring constraint NoVoice is increased by 1 and the weight of the loser-preferring constraint IdentVoice is decreased by 1, just as in the case of the original Perceptron. The crucial difference is that the weight of the constraint IdentVoiceOnset is left unchanged in order to prevent it from turning negative, despite the fact that the constraint is loser-preferring.

(12) *Update by the truncated Perceptron:*

$$\begin{array}{r} \textsc{IdentVoiceOnset} \\ \textsc{IdentVoice} \\ \textsc{NoVoice} \end{array} \begin{bmatrix} 0 \\ 7 \\ 3 \end{bmatrix} \xrightarrow{(/\text{da}/,\ [\text{ta}],\ \cancel{[\text{da}]})} \begin{bmatrix} 0 \\ 6 \\ 4 \end{bmatrix}$$

Crucially, the update (12) by the truncated Perceptron can be analyzed as the sequence (13) of two updates by the original Perceptron. At the first update (13a), the original Perceptron is fed the piece of data (/da/, [ta], ~~[da]~~) as in (10). Thus, in particular, the weight of the loser-preferring constraint IdentVoiceOnset is demoted to −1.

(13) *Sequence of two updates by the original Perceptron:*

$$\begin{array}{r} \textsc{IdentVoiceOnset} \\ \textsc{IdentVoice} \\ \textsc{NoVoice} \end{array} \begin{bmatrix} 0 \\ 7 \\ 3 \end{bmatrix} \underbrace{\xrightarrow{(/\text{da}/,\ [\text{ta}],\ \cancel{[\text{da}]})} \begin{bmatrix} -1 \\ 6 \\ 4 \end{bmatrix}}_{\text{a.}} \underbrace{\xrightarrow{(x,\ y,\ \cancel{z})} \begin{bmatrix} 0 \\ 6 \\ 4 \end{bmatrix}}_{\text{b.}}$$

Immediately afterwards, the original Perceptron is fed with the "dummy" piece of data described in (14). This piece of data consists of the underlying form x together with the corresponding winner candidate y and the loser candidate $\cancel{z}$. The only constraint which distinguishes between these two candidates is IdentVoiceOnset, which prefers the winner. There are no loser-preferring constraints. In other words, the violation differences corresponding to this triplet $(x, y, \cancel{z})$ are all null apart from the one corresponding to the constraint IdentVoiceOnset which is equal to +1. Condition (2) fails: the right hand side is null (because there are no loser-preferring constraints) and the left-hand side is negative (because IdentVoiceOnset is the only winner-preferring constraint, and its current weight −1 is negative). The original Perceptron thus performs the update (13b): the weight of the winner-preferring constraint IdentVoiceOnset is increased by 1 back to zero, and no other weights are modified.

(14)

Input: x	IdentVoiceOnset $\theta = -1$	IdentVoice $\theta = 6$	NoVoice $\theta = 4$
a. y			
b. ☞ z	$*$(W)		

In the end, the final weights after the two updates (13) by the original Perceptron are identical to the final weights after the single update (12) by the truncated Perceptron.

3.4 *Convergence of the truncated Perceptron reweighing rule*

The analysis of this specific example extends to the general case. Any update according to the truncated Perceptron reweighing rule (9) can be analyzed as a sequence of updates according to the original Perceptron reweighing rule (5), namely the update triggered by the actual piece of data followed by some updates triggered by *dummy* data which undo the illicit demotions that yielded negative weights. These dummy data have a winner-preferring constraint but no loser-preferring constraints. In other words, their constraint violation differences are all null apart for one, which is equal to +1. If the training data are consistent with some HG grammar, the training plus the dummy data are consistent as well (see Appendix A.5). Of course, these dummy data have no phonological meaning. Indeed, I am *not* suggesting that the set of phonological forms should be extended with these dummy data. These artificial data only play a role in the analysis (not in the simulations) of the truncated Perceptron.

Let me take stock. The convergence Theorem 1 for the original Perceptron ensures convergence whenever the training data are consistent. A run of the truncated Perceptron can be analyzed as a run of the original Perceptron on the training data extended with dummy data which undo forbidden reweighing. Furthermore, consistency of the original training data guarantees consistency of the extended data. The convergence Theorem 1 for the original Perceptron thus yields the analogous convergence Theorem 2 for the truncated Perceptron. Appendix A.6 formalizes the reasoning sketched above into a proof. See Magri (to appear) for more discussion of the error bound (15).

Theorem 2 *Let the set of dummy data consist of underlying/winner/loser form triplets whose violation differences are all equal to zero apart from one which is equal to* $+1$*. The HG error-driven learner (1) with the HG update condition (3) and the* truncated *Perceptron reweighing rule (9) converges: the number of errors is bounded by*

$$\left(\frac{\textit{radius of the training data}}{\textit{margin of the training plus dummy data}}\right)^2 \tag{15}$$

when the training set consists of underlying/winner/loser form triplets which are all consistent with some HG grammar and have bounded violation differences.

The error bound (15) for the truncated Perceptron only differs from the error bound (6) for the original Perceptron because the latter has the margin of only the training data at the denominator while the former has the margin of the training plus dummy data. Let me comment on this difference. The margin of a training set quantifies its degree of consistency. Intuitively, extending a training set with additional data can only shrink the degree of consistency (any grammar consistent with the extended training set is also consistent with the original one, but not vice versa; see Appendix A.3). Hence, the margin of the original training set extended with the dummy data which appears in the error bound (15) for the truncated Perceptron is equal to or smaller than the margin of just the original training set which appears in the error bound (6) for the original Perceptron. The error bound (15) for the truncated Perceptron is therefore worse than (namely, at least as large as) the error bound (6) for the original Perceptron. The difference between the two margins quantifies the price that needs to be paid for HG's assumption (7) of non-negative weights.

4 EXTENSION TO THE STOCHASTIC IMPLEMENTATION AND THE NOISY SETTING

This section extends the analysis of the truncated Perceptron to the stochastic implementation and the noisy learning setting.

4.1 *Stochastic implementation*

The implementation of error-driven learning considered so far is called *deterministic*, to distinguish it from the *stochastic* implementation (Boersma 1997, 1998; Boersma and Hayes 2001; Coetzee and Pater 2008, 2011; Coetzee and Kawahara 2013; Boersma and Pater to appear; Jarosz 2013). Intuitively, the latter differs because the current piece of data is compared not with the current grammar but with a variant thereof sampled from a neighborhood of the current grammar. This intuition can be formalized as follows. At step (1b), the deterministic HG error-driven learner checks whether the current weights $\theta_1, \ldots, \theta_n$ satisfy the update condition (2) or (3), depending on whether the constraints are binary or possibly gradient. The only innovation of the stochastic implementation is that this update condition is checked not for the current weights $\theta_1, \ldots, \theta_n$ but for the *stochastic weights* $\theta_1 + \epsilon_1, \ldots, \theta_n + \epsilon_n$, obtained by adding to the current weights certain values $\epsilon_1, \ldots, \epsilon_n$ sampled independently from each other according to the same underlying distribution. In other words, the learner checks the *stochastic update conditions* (16) or (17), depending on whether the constraints are binary or possibly gradient.

(16) $$\sum_{h \in W} (\theta_h + \epsilon_h) > \sum_{k \in L} (\theta_k + \epsilon_k)$$

(17) $$\sum_{k=1}^{n} \Big(\underbrace{C_k(x, z) - C_k(x, y)}_{\text{violation difference}} \Big) (\theta_k + \epsilon_k) > 0$$

These stochastic values ϵ_k are usually sampled according to a gaussian distribution with zero mean and small variance (Boersma 1997, 1998; Boersma and Hayes 2001). Since the tails of the gaussian distribution decrease exponentially fast, these stochastic values are bounded *with high probability* between some thresholds $-\Delta$ and $+\Delta$. From an analytical perspective, it is nonetheless convenient to assume they are *deterministically* bounded, namely sampled according to a distribution concentrated between $-\Delta$ and $+\Delta$. The analyses carry over with high probability to the gaussian distribution. The algorithm (1) with the update condition (16)/(17) at step (1b) is called the HG *stochastic* error-driven learner[5] (Boersma 1997, 1998; Boersma and Hayes 2001; Coetzee and Pater 2008, 2011; Coetzee and Kawahara 2013; Boersma and Pater to appear, Jarosz 2013).

For simplicity, assume that all constraints are binary (the reasoning extends to the general case). The stochastic update condition (16) can be rewritten as in (18), where the value ϵ on the right hand side is the sum of those stochastic values $\epsilon_1, \epsilon_2, \dots$ which correspond to the loser-preferring constraints minus the sum of those stochastic values which instead correspond to the winner-preferring constraints.[6]

(18) $$\sum_{h \in W} \theta_h - \sum_{k \in L} \theta_k > \epsilon$$

The stochastic update condition (18) is thus almost identical to the deterministic update condition (2), repeated in (19) with all the terms rearranged on the left. The only difference is that zero on the right hand side of (19) is replaced by ϵ in (18). Yet, ϵ cannot be much different from zero, since it is the sum of numbers sampled between $-\Delta$ and $+\Delta$.

(19) $$\sum_{h \in W} \theta_h - \sum_{k \in L} \theta_k > 0$$

Since the deterministic and stochastic implementations only differ for the update conditions and since these conditions differ only minimally, the convergence Theorem 1 for the original deterministic Perceptron extends to the stochastic variant. Based on this reasoning, Boersma and Pater (to appear) obtain the convergence guarantees for the stochastic *original* Perceptron summarized in Theorem 3. The error bound (20) is the sum of two terms. The first term (20a) coincides with the error bound (6) for the deterministic HG learner. The second term (20b) thus quantifies the number of additional errors due to the stochastic implementation.

Theorem 3 *Assume that the stochastic values $\epsilon_1, \dots, \epsilon_n$ of the n constraints are sampled independently in between $-\Delta$ and $+\Delta$ for some con-*

[5] It is called instead the *Noisy HG-GLA* in Boersma and Pater (to appear). As explained in footnote 2, I prefer not to use the acronym "GLA" in the context of HG. Furthermore, I prefer "stochastic" over "noisy", in order to avoid any confusion between the stochastic implementation considered here and the noisy learning setting considered in Subsection 4.2.

[6] Namely: $\epsilon = \sum_{k \in L} \epsilon_k - \sum_{h \in W} \epsilon_h$.

stant $\Delta \geq 0$. The HG error-driven learner with the stochastic update condition and the original Perceptron reweighing rule converges: the number of errors is bounded by

$$(20)\quad \underbrace{\left(\frac{\textit{radius of training data}}{\textit{margin of training data}}\right)^2}_{\text{(a)}} + \underbrace{2n\Delta\frac{\begin{array}{c}\textit{largest absolute value}\\ \textit{of violation differences}\end{array}}{\left(\textit{margin of training data}\right)^2}}_{\text{(b)}}$$

when the training set consists of underlying/winner/loser form triplets which are all consistent with some HG grammar and have bounded violation differences.

By reasoning as in Subsection 3.3, this result extends to the stochastic *truncated* Perceptron, yielding the following Theorem 4. Again, the only difference between the two error bounds (20) and (21) for the original and the truncated Perceptron is that the denominator of the former has the margin of the training data while the denominator of the latter has the margin of the training plus dummy data.

Theorem 4 *Let the set of dummy data consist of underlying/winner/loser form triplets whose violation differences are all equal to zero apart from one which is equal to* $+1$*. Assume that the stochastic values* $\epsilon_1, \dots, \epsilon_n$ *of the* n *constraints are sampled in between* $-\Delta$ *and* $+\Delta$ *for some constant* $\Delta \geq 0$*. The HG error-driven learner (1) with the stochastic update condition and the* truncated *Perceptron reweighing rule converges: the number of errors is bounded by*

$$(21)\quad \left(\frac{\textit{radius of training data}}{\begin{array}{c}\textit{margin of training}\\ \textit{plus dummy data}\end{array}}\right)^2 + 2n\Delta\frac{\begin{array}{c}\textit{largest absolute value}\\ \textit{of violation differences}\end{array}}{\left(\begin{array}{c}\textit{margin of training}\\ \textit{plus dummy data}\end{array}\right)^2}$$

when the training set consists of underlying/winner/loser form triplets which are all consistent with some HG grammar and have bounded violation differences.

4.2 *Noisy learning setting*

A realistic learning setting needs to allow for the possibility that the (possibly infinite) sequence of *pristine* training data generated by some target grammar has been interspersed with data *corrupted* by transmission noise or production errors (Gibson and Wexler 1994, p. 410;

Frank and Kapur 1996, p. 625; Boersma and Hayes 2001, pp. 66–67; Bíró 2006, among many others). No assumptions are made on the corrupted data, apart from there being only a finite number of them.[7] The classical error bound for the Perceptron algorithm in this noisy learning setting is due to Freund and Schapire (1999) (building on Klasner and Simon 1995; see also Mohri *et al.* 2012, ch. 7 for a textbook treatment). The shape of their bound is recalled in (22). The precise definition of the quantity which appears as the second term in the numerator is somewhat involved and therefore relegated to Appendix A.7. What is crucial is that this quantity is null when there are no corrupted training data and grows with the number of corrupted data. The error bound (22) differs from the error bound (6) for the noise-free setting because of this additional quantity, which thus quantifies the additional number of errors due to the corrupted training data. Subsequent improvements of Freund and Shapire's error bound (Shalev-Shwartz and Singer 2005; Mohri and Rostamizadeh 2013) do not alter its basic shape (22).

Theorem 5 *Consider the HG error-driven learner with the deterministic update condition*[8] *and the original Perceptron reweighing rule. Suppose it is trained on a (possibly infinite) sequence of* pristine *training data consisting of underlying/winner/loser form triplets which are all consistent with some HG grammar and have bounded violation differences. Suppose that this sequence is interspersed with a finite number of arbitrary* corrupted *data. The number of errors made by the learner on this corrupted training sequence is bounded by:*

$$(22)\quad \left(\frac{\begin{array}{c}\textit{radius of the pristine}\\ \textit{plus corrupted data}\end{array} + \begin{array}{c}\textit{a quantity which depends}\\ \textit{on the corrupted data}\end{array}}{\textit{margin of the pristine data}}\right)^2$$

[7] Indeed, if an infinite number of corrupted data were allowed, the worst case number of errors would always be infinite: whenever the learner rests on a current hypothesis, we can prompt it to perform yet another update by maliciously crafting an appropriate piece of corrupted data.

[8] It is only for simplicity that the analysis of the noisy learning setting is limited to the deterministic implementation. Theorems 3 and 5 can be easily combined, yielding an error bound for the HG stochastic learner in the noisy setting.

By reasoning as in Subsection 3.3, this result extends to the truncated Perceptron, yielding the following Theorem 6. Again, the only difference between the two error bounds (22) and (23) for the original and the truncated Perceptron is that the denominator of the former has the margin of the pristine training data while the denominator of the latter has the margin of the pristine training data plus the dummy data.

Theorem 6 *Let the set of dummy data consist of underlying/winner/loser form triplets whose violation differences are all equal to zero apart from one which is equal to* $+1$. *Consider the HG error-driven learner with the deterministic update condition and the* truncated *Perceptron reweighing rule. Suppose it is trained on a (possibly infinite) sequence of* pristine *training data consisting of underlying/winner/loser form triplets which are all consistent with some HG grammar and have bounded violation differences. Suppose that this sequence is interspersed with a finite number of arbitrary* corrupted *data. The number of errors made by the learner on this training sequence can be bounded by:*

(23)
$$\left(\frac{\begin{array}{c}\textit{radius of the pristine}\\ \textit{plus corrupted data}\end{array} + \begin{array}{c}\textit{a quantity which depends}\\ \textit{on the corrupted data}\end{array}}{\textit{margin of the pristine plus dummy data}}\right)^2$$

5 CONCLUSIONS

The current HG error-driven learning literature has adopted the Perceptron reweighing rule. Yet, this reweighing rule is not suited to HG, as it does not guarantee non-negativity of the weights. I have thus considered a variant whereby the updates are "truncated" at zero, enforcing non-negativity of the weights in a principled way. A run of the truncated Perceptron can be analyzed as a run of the original Perceptron on the same training sequence interspersed with dummy data used to "undo" the truncated updates. Convergence guarantees for the original Perceptron (Theorem 1), its stochastic implementation (Theorem 3), and its noise robustness (Theorem 5) thus extend to the truncated variant (Theorems 2, 4, and 6). This observation provides the first constraint-independent convergence guarantees for an HG error-driven learner consistent with HG's restriction to non-negative weights.

A APPENDICES

A.1 *Representing the training data as EWCs*

At each iteration, the error-driven learner (1) processes a piece of data which consists of a certain winner candidate y and a certain loser candidate $\not{z}$ for a certain underlying form x. Denote by a_k the difference between the number $C_k(x, \not{z})$ of violations assigned by constraint C_k to the loser mapping minus the number $C_k(x, y)$ of violations assigned to the winner-mapping, namely $a_k = C_k(x, \not{z}) - C_k(x, y)$. Collect these violation differences corresponding to the constraints $C_1, \ldots, C_n$ into a vector $\mathbf{a} = (a_1, \ldots, a_n)$, called an *elementary weighting condition* (EWC), in analogy with Prince's 2002 *elementary ranking conditions* in Optimality Theory. The consistency condition (3) between a weight vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ and an underlying/winner/loser form triplet is only stated in terms of the violation differences, not in terms of the actual numbers of constraint violations. It can thus be restated as in (24a) in terms of the EWC $\mathbf{a} = (a_1, \ldots, a_n)$ corresponding to the data triplet. Also the original and the truncated Perceptron reweighing rules (5) and (9) are only stated in terms of violation differences, and can thus be restated as in (24b) and (24c) in terms of EWCs.

(24) a. $$\sum_{k=1}^{n} a_k \theta_k > 0$$

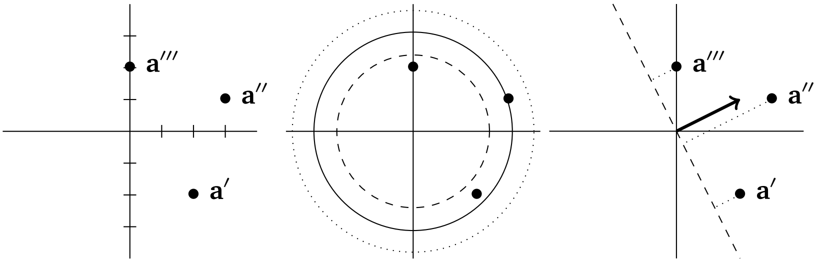
b. $\theta_k \leftarrow \theta_k + a_k$

c. $$\theta_k \leftarrow \begin{cases} \theta_k + a_k & \text{if } \theta_k + a_k \geq 0 \\ \theta_k & \text{otherwise} \end{cases}$$

In conclusion, the piece of training data $(x, y, \not{z})$ fed to the learner at step (1a) can be represented as an EWC. Throughout this appendix, I thus assume that the HG learner is trained on a sequence of EWCs sampled from a certain *training set* $\mathbf{A}$ of EWCs.

A.2 *Geometric definition of the radius*

Suppose there are only $n = 2$ constraints C_1 and C_2. A generic EWC thus has the shape $\mathbf{a} = (a_1, a_2)$, where a_1 and a_2 are the violation differences corresponding to the two constraints C_1 and C_2, respectively. The EWC can thus be represented with a point in the cartesian plane, through the convention that the horizontal axis corresponds to constraint C_1 and the vertical axis corresponds to constraint C_2. To illustrate, the



a. Example of EWCs b. Example of radius c. Example of margin

Figure 1: Geometric illustration of EWCs (left), radius (center) and margin (right).

EWC set $\mathbf{A} = \{\mathbf{a}', \mathbf{a}'', \mathbf{a}''\}$ consisting of the three EWCs $\mathbf{a}' = (2, -2)$, $\mathbf{a}'' = (3, 1)$, and $\mathbf{a}''' = (0, 2)$ can be represented as in Figure 1a.

Consider now various circles of different radiuses centered in the origin. The radius could be too small, so that the corresponding circle fails at containing all EWCs in $\mathbf{A}$, as in the case of the dashed circle in Figure 1b. Or the radius could be too large, so that the corresponding circle contains all EWCs with some slack, as in the case of the dotted circle. Or the radius could coincide with the distance from the origin of the EWC furthest away, so that the corresponding circle contains all EWCs without any slack, as in the case of the solid circle. The radius of the latter solid circle in Figure 1b is univocally determined. It is called the *radius of the EWC set* $\mathbf{A}$ and denoted by $\rho(\mathbf{A})$. The extension from $n = 2$ to an arbitrary number n of constraints is conceptually straightforward. The analytic definition of the radius for an arbitrary number n of constraints is provided in (25a) in Appendix A.4.

A.3 *Geometric definition of the margin*

With only $n = 2$ constraints, a generic weight vector has the shape $\boldsymbol{\theta} = (\theta_1, \theta_2)$: it consists of the weights θ_1 and θ_2 of the two constraints C_1 and C_2. The corresponding *decision line* is the line through the origin which is perpendicular to the arrow which starts at the origin and ends at the point whose horizontal and vertical coordinates are θ_1 and θ_2 respectively. To illustrate, the decision line corresponding to the weight vector $\boldsymbol{\theta} = (2, 1)$ is represented by the dashed line in Figure 1c. The decision line splits the plane into two half planes, one of which contains the arrow. The consistency condition (24a) between a weight vector and an EWC says that the EWC lies in the half-plane which contains the arrow which represents the weight vector. To illustrate, Figure 1c shows that the weight vector considered is consistent with the EWC set $\mathbf{A} = \{\mathbf{a}', \mathbf{a}'', \mathbf{a}'''\}$, because all three EWCs lie in the half-plane containing the arrow.

The *distance* of an EWC $\mathbf{a}$ from the decision line is the length of the segment which starts at $\mathbf{a}$ and falls perpendicularly on the decision line, represented by the dotted segments in 1c. This distance can be interpreted as the "degree of consistency" of the EWC with (the decision line corresponding to) the weight vector. Thus, although the weight vector plotted in Figure 1c is consistent with both EWCs $\mathbf{a}'$ and $\mathbf{a}''$, the former EWC is closer to the decision line and thus has a smaller degree of consistency than the latter. Indeed, a small perturbation of the weights slightly rotates the decision line and might affect consistency with the closer $\mathbf{a}'$ but not with $\mathbf{a}''$. Since we are interested in worst-case analyses, we focus on the most "dangerous" EWC in the EWC set $\mathbf{A}$, namely the one which is closest to the decision line and thus has the smallest degree of consistency. The distance of that EWC from the decision line is called the *margin* of the EWC set $\mathbf{A}$ with respect to the weight vector $\boldsymbol{\theta}$, and is denoted by $\mu(\mathbf{A}, \boldsymbol{\theta})$. To illustrate, the margin of the EWC set $\mathbf{A} = \{\mathbf{a}', \mathbf{a}'', \mathbf{a}'''\}$ relative to the decision line represented by the dashed line in Figure 1c is the distance of either EWCs $\mathbf{a}'$ or $\mathbf{a}'''$.

Different weight vectors induce different decision lines that in turn differ because of their distances from the various EWCs. Among all weight vectors consistent with the EWC set, consider a weight vector $\widehat{\boldsymbol{\theta}}$ whose decision line achieves the largest distance from the closest EWC, namely whose margin $\mu(\mathbf{A}, \widehat{\boldsymbol{\theta}})$ is at least as large as the margin $\mu(\mathbf{A}, \boldsymbol{\theta})$ relative to any other weight vector $\boldsymbol{\theta}$. The margin of any such *optimal* weight vector is called the *margin* of the EWC set $\mathbf{A}$ and is denoted by $\mu(\mathbf{A})$. As is clear from this geometric definition, all optimal weight vectors correspond to the same decision line, which is therefore unique. The extension from $n = 2$ to an arbitrary number n of constraints is conceptually straightforward. The analytic definition of the margin for an arbitrary number n of constraints is provided in (25b) in Appendix A.4.

A.4 *Analytical expression of the radius and the margin*

Let $\langle \cdot, \cdot \rangle$ be the *Euclidean scalar product,* defined by $\langle \mathbf{v}, \mathbf{w} \rangle = \sum_{i=1}^{n} v_i w_i$ for any pair of vectors $\mathbf{v} = (v_1, \dots, v_n)$ and $\mathbf{w} = (w_1, \dots, w_n)$. Let $\| \cdot \|$ be the *Euclidean norm,* defined by $||\mathbf{v}||^2 = \langle \mathbf{v}, \mathbf{v} \rangle = \sum_{i=1}^{n} v_i^2$. The consistency condition (24a) between a weight vector $\boldsymbol{\theta}$ and an EWC $\mathbf{a}$ can thus be rewritten as the condition $\langle \boldsymbol{\theta}, \mathbf{a} \rangle > 0$. The radius $\rho(\mathbf{A})$ and the margin $\mu(\mathbf{A})$ of a finite EWC set $\mathbf{A}$, which were defined geometrically in Appendices A.2 and A.3, can now be expressed analytically for an arbitrary number n of constraints as in (25).

(25) a. $\rho(\mathbf{A}) = \max_{\mathbf{a} \in \mathbf{A}} ||\mathbf{a}||$

b. $\mu(\mathbf{A}) = \max_{\boldsymbol{\theta} \neq \mathbf{0}} \mu(\boldsymbol{\theta}, \mathbf{A})$ where $\mu(\mathbf{A}, \boldsymbol{\theta}) = \min_{\mathbf{a} \in \mathbf{A}} \frac{\langle \boldsymbol{\theta}, \mathbf{a} \rangle}{||\boldsymbol{\theta}||}$

The assumption that the set $\mathbf{A}$ is finite ensures that the maxima over $\mathbf{A}$ are well defined. This assumption is not restrictive. In fact, all the theorems considered in the paper assume that the training set consists of underlying/winner/loser form triplets with bounded violation differences. Since the violation differences are integers, this boundedness assumption is equivalent to the assumption that the EWC set $\mathbf{A}$ corresponding to the training set is finite.

A.5 *Consistency of the training plus dummy data*

The analysis of the truncated Perceptron sketched in Section 3 relies on the notion of *dummy data.* These are underlying/winner/loser form triplets which have a unique non-zero constraint violation difference, which is equal to $+1$. The set of EWCs corresponding to these dummy data will be denoted by $\mathbf{E}$. Thus, an EWC $\mathbf{e}$ in $\mathbf{E}$ is a vector which has a unique non-zero component, which is equal to $+1$.

Denote by $\mathbf{A}$ the set of EWCs corresponding to the underlying/winner/loser form triplets the learner is trained on. Suppose that this training set is consistent with the HG grammar corresponding to some weight vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$. Can I conclude that the set $\mathbf{A} \cup \mathbf{E}$ obtained by extending the training set $\mathbf{A}$ with the dummy data $\mathbf{E}$ is consistent with $\boldsymbol{\theta}$ as well? Since each dummy EWC $\mathbf{e}$ has no negative components and the weight vector $\boldsymbol{\theta}$ has nonnegative components, that is indeed the case as long as all the weights θ_k are all different from zero, namely not only non-negative but actually strictly positive. If that is not the case, then consistency with the dummy EWC set $\mathbf{E}$ might fail. For instance, the dummy EWC $\mathbf{e} = (1, 0, \ldots, 0)$ (whose unique non-null component corresponds to constraint C_1) is not consistent with a weight vector $\boldsymbol{\theta}$ which assigns to constraint C_1 a null weight $\theta_1 = 0$ (because $\langle \boldsymbol{\theta}, \mathbf{e} \rangle = 0 \not> 0$). Yet, the following lemma guarantees that a consistent EWC set **A** is always consistent with weights which are strictly positive (namely neither negative nor equal to zero), as weights which are equal to zero can be slightly increased without compromising consistency. This lemma will be used below for the proof of the convergence Theorem 2 for the truncated Perceptron.

Lemma 1 *A finite set* **A** *of EWCs consistent with some HG grammar is in particular consistent with an HG grammar corresponding to weights which are all strictly positive.*

Proof. The hypothesis that **A** is consistent means that there exists a weight vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ of non-negative weights $\theta_k \geq 0$ such that $\langle \boldsymbol{\theta}, \mathbf{a} \rangle > 0$ for every EWC **a** in **A**. If all the weights happen to be strictly positive (i.e., $\theta_k > 0$), then the claim is proven. Thus, assume that some weights are equal to zero. Let Ω be the set of those indices k such that the corresponding weight θ_k is strictly positive and let $\overline{\Omega}$ be its complement, as defined in (26).

(26) $\Omega = \left\{ k \in \{1, \ldots, n\} \,\middle|\, \theta_k > 0 \right\}$ $\qquad$ $\overline{\Omega} = \left\{ k \in \{1, \ldots, n\} \,\middle|\, \theta_k = 0 \right\}$

I will now construct another weight vector $\widehat{\boldsymbol{\theta}} = (\widehat{\theta}_1, \ldots, \widehat{\theta}_n)$ which has all positive weights $\widehat{\theta}_k > 0$ and furthermore is consistent with **A** as well. Let the constants A and B be defined as in (27), which makes sense because of the assumption that the training EWC set **A** is finite. The constant A is strictly positive, because the original weight vector $\boldsymbol{\theta}$ is consistent with every EWC **a** in **A**. The constant B is instead strictly negative, because at least one EWC needs to have a negative entry (otherwise the claim is trivial).

(27) a. $A = \min_{\mathbf{a} \in \mathbf{A}} \langle \boldsymbol{\theta}, \mathbf{a} \rangle$ $\qquad$ b. $B = \min_{\mathbf{a} = (a_1, \ldots, a_n) \in \mathbf{A}} \min_k a_k$

Define the new weight vector $\widehat{\boldsymbol{\theta}} = (\widehat{\theta}_1, \ldots, \widehat{\theta}_n)$ as in (28). The weights thus defined are all strictly positive as desired, because the constant A is strictly positive and the constant B is strictly negative. In general, A is a small value and $|B|$ is a large value. Thus we have effectively only slightly perturbed the original weight vector $\boldsymbol{\theta}$ by replacing its null weights with a small positive value.

(28)
$$\widehat{\theta}_k = \begin{cases} \theta_k & \text{if } k \in \Omega \\ -\dfrac{A}{2(n-1)B} & \text{if } k \in \overline{\Omega} \end{cases}$$

The scalar product between the perturbed weight vector $\widehat{\boldsymbol{\theta}}$ and an arbitrary EWC **a** in **A** can be computed as in (29), which shows that $\widehat{\boldsymbol{\theta}}$ is consistent with **a**.

$$
\begin{aligned}
(29) \quad \langle \widehat{\boldsymbol{\theta}}, \mathbf{a} \rangle &= \sum_{k\in\Omega} \widehat{\theta}_k a_k + \sum_{k\in\overline{\Omega}} \widehat{\theta}_k a_k \\
&\overset{(a)}{=} \sum_{k\in\Omega} \theta_k a_k - \sum_{k\in\overline{\Omega}} \frac{A}{2(n-1)B} a_k \\
&\overset{(b)}{=} \sum_{k\in\Omega} \theta_k a_k + \sum_{k\in\overline{\Omega}} \theta_k a_k - \sum_{k\in\overline{\Omega}} \frac{A}{2(n-1)B} a_k \\
&= \langle \boldsymbol{\theta}, \mathbf{a} \rangle - \sum_{k\in\overline{\Omega}} \frac{A}{2(n-1)B} a_k \\
&\overset{(c)}{\geq} A - \sum_{k\in\overline{\Omega}} \frac{A}{2(n-1)B} a_k \\
&\overset{(d)}{\geq} A - \sum_{k\in\overline{\Omega}} \frac{A}{2(n-1)B} B \\
&\geq A - \sum_{k\in\overline{\Omega}} \frac{A}{2(n-1)} \\
&\overset{(e)}{\geq} A - \frac{A}{2} \\
&> 0
\end{aligned}
$$

In step (29a), I have used the position (28). In step (29b), I have added the quantity $\sum_{k\in\overline{\Omega}} \theta_k a_k$, which is null because the weights θ_k corresponding to indices $k \in \overline{\Omega}$ are all null. In step (29c), I have lower-bounded by replacing $\langle \boldsymbol{\theta}, \mathbf{a} \rangle$ with the smallest possible value A. In step (29d), I have lower-bounded by replacing a_k with its smallest possible value B (this step is licit, because a_k is multiplied by a positive coefficient, since B is negative). In step (29e), I have used the fact that the original weight vector $\boldsymbol{\theta}$ can contain at most $n-1$ null weights (at least one weight needs to be non-null in order for $\boldsymbol{\theta}$ to yield a strictly positive scalar product with the EWCs in **A**), so that the sum over $\overline{\Omega}$ has at most $n-1$ terms. ■

A.6 *Proof of the convergence Theorem 2 for the truncated Perceptron*

Using the preceding lemma, I can now straightforwardly formalize the reasoning sketched in Subsection 3.2 into a proof of the convergence Theorem 2 for the truncated Perceptron, restated below in terms of EWCs.

Theorem 2. *Let* $\mathbf{E}$ *be the set of the* dummy *EWCs, whose components are all zeros but for one component which is instead equal to* $+1$*. The HG error-driven learner with the deterministic update condition (24a) and the* truncated *Perceptron reweighing rule (24c) converges: when trained on a finite EWC set* $\mathbf{A}$ *consistent with some HG grammar, the number of errors is bounded by*

$$(30)\quad \left(\frac{\rho(\mathbf{A})}{\mu(\mathbf{A}\cup\mathbf{E})}\right)^2$$

where $\rho(\mathbf{A})$ *is the radius of the training set* $\mathbf{A}$ *and* $\mu(\mathbf{A}\cup\mathbf{E})$ *is the margin of the training set* $\mathbf{A}$ *extended with the dummy set* $\mathbf{E}$.

Proof. By reasoning as in Subsection 3.2, any run of the HG error-driven learner with the truncated Perceptron reweighing rule on a training EWC set $\mathbf{A}$ can be mimicked with a run of the algorithm with the original Perceptron reweighing rule on the extended EWC set $\mathbf{A}\cup\mathbf{E}$. In fact, suppose that the truncated Perceptron leaves a weight θ_k at zero while the original Perceptron demotes it down to, say, -5. Then, the original Perceptron can be forced to bring it back to zero by feeding it five times with the EWC in $\mathbf{E}$ which has all components equal to zero but for the kth component which is equal to 1. In other words, the EWCs in $\mathbf{E}$ play the role of the "dummy data" considered in Subsection 3.2. The worst-case number of errors $T_{\text{truncated}}(\mathbf{A})$ made by the truncated Perceptron on the training set $\mathbf{A}$ can thus be bounded as in (31) in terms of the number of errors $T_{\text{original}}(\mathbf{A}\cup\mathbf{E})$ made by the original Perceptron on the extended training set $\mathbf{A}\cup\mathbf{E}$.

$$(31)\quad T_{\text{truncated}}(\mathbf{A}) \leq T_{\text{original}}(\mathbf{A}\cup\mathbf{E})$$

Since the training set $\mathbf{A}$ is finite and consistent with some HG grammar, lemma 1 ensures that it is in particular consistent with a weight vector $\boldsymbol{\theta}$ of strictly positive weights. Since any vector of strictly positive weights is consistent with the EWCs in $\mathbf{E}$, I conclude that this weight vector $\boldsymbol{\theta}$ is consistent with the extended training set $\mathbf{A}\cup\mathbf{E}$. The Perceptron convergence Theorem 1 thus applies, ensuring that the worst-case number of errors $T_{\text{original}}(\mathbf{A}\cup\mathbf{E})$ made by the original Perceptron on the extended EWC set $\mathbf{A}\cup\mathbf{E}$ can be bounded in terms of its radius and margin as in (32).

$$(32)\quad T_{\text{original}}(\mathbf{A}\cup\mathbf{E}) \leq \left(\frac{\rho(\mathbf{A}\cup\mathbf{E})}{\mu(\mathbf{A}\cup\mathbf{E})}\right)^2$$

The radius of the extended training set $\mathbf{A} \cup \mathbf{E}$ is equal to the radius of the original training set $\mathbf{A}$, as computed in (33). In the first equality, I have used the definition (25a) of the radius. In the second equality, I have used the fact that the vectors $\mathbf{e} \in \mathbf{E}$ are unit vectors, namely $\|\mathbf{e}\| = 1$. Finally, in the third equality, I have used the fact that each EWC $\mathbf{a} \in \mathbf{A}$ has integer components, so that $\|\mathbf{a}\| \geq 1$.

(33)
$$\begin{aligned}\rho(\mathbf{A} \cup \mathbf{E}) &= \max\left\{\max_{\mathbf{a}\in\mathbf{A}}\|\mathbf{a}\|,\ \max_{\mathbf{e}\in\mathbf{E}}\|\mathbf{e}\|\right\} = \max\left\{\max_{\mathbf{a}\in\mathbf{A}}\|\mathbf{a}\|,\ 1\right\}\\ &= \max_{\mathbf{a}\in\mathbf{A}}\|\mathbf{a}\| = \rho(\mathbf{A})\end{aligned}$$

The claim follows by combining (31), (32), and (33). ■

The identity (33) shows that the radius $\rho(\mathbf{A} \cup \mathbf{E})$ of the extended EWC set $\mathbf{A} \cup \mathbf{E}$ coincides with the radius $\rho(\mathbf{A})$ of the original EWC set $\mathbf{A}$. This is not true for the margin: the margin $\mu(\mathbf{A} \cup \mathbf{E})$ of the extended EWC set can be smaller than the margin $\mu(\mathbf{A})$ of the original EWC set.

A.7 *Error-bound for the noisy learning setting*

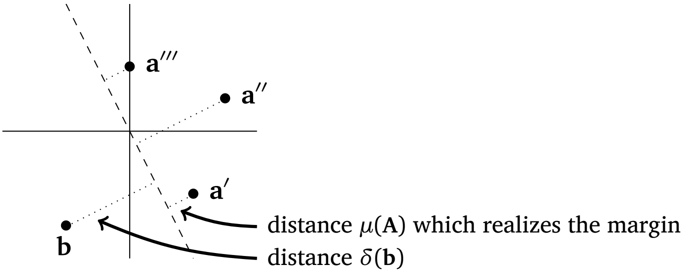
Theorem 5 from Subsection 4.2 provides the approximate expression (22) of the error bound for the HG error-driven learner in the noisy learning setting. The precise formulation of the error bound is provided in (34).

Theorem 5. *Consider the HG error-driven learner with the deterministic update condition (24a) and the original Perceptron reweighing rule (24b). Assume it is trained on a sequence of EWCs sampled from two EWC sets* $\mathbf{A}$ *and* $\mathbf{B}$. *The EWCs of* $\mathbf{A}$ *are called* pristine *because they are consistent with some HG grammar with margin* $\mu(\mathbf{A})$. *The EWCs of* $\mathbf{B}$ *are called* corrupted *because each of them is inconsistent with the EWCs in* $\mathbf{A}$. *Assume that the set* $\mathbf{A}$ *of pristine EWCs is finite and that the training sequence contains only a finite number of corrupted EWCs from* $\mathbf{B}$. *The number of errors made by the learner on this training sequence is at most*

(34)
$$\left(\frac{\rho(\mathbf{A} \cup \mathbf{B}) + \sqrt{\sum_{\mathbf{b}\in\mathbf{B}} n(\mathbf{b})\big(\mu(\mathbf{A}) + \delta(\mathbf{b})\big)^2}}{\mu(\mathbf{A})}\right)^2$$

where $\rho(\mathbf{A} \cup \mathbf{B})$ *is the radius of the pristine data* $\mathbf{A}$ *plus the corrupted data* $\mathbf{B}$, $\mu(\mathbf{A})$ *is the margin of the pristine data* $\mathbf{A}$, $n(\mathbf{b})$ *is the number of times*

Figure 2: Illustration of Theorem 5



that the corrupted piece of data **b** *has been fed to the learner in the training sequence, and* $\delta(\mathbf{b})$ *is the distance of the corrupted piece of data* **b** *from the decision surface corresponding to the weight vector which realizes the margin of the pristine data.*

The theorem can be illustrated as follows. Suppose that there are only $n = 2$ constraints and that the set of pristine EWCs is $\mathbf{A} = \{\mathbf{a}', \mathbf{a}'', \mathbf{a}'''\}$ plotted in Figure 2. The decision line which realizes the margin of these pristine data is represented by the dashed line. The margin is the distance $\mu(\mathbf{A})$ of the closest EWC $\mathbf{a}'$ from the dashed line. The EWC **b** is corrupted because inconsistent with the pristine data (it sits in the opposite half plane). The distance of this corrupted piece of data **b** from the decision line which realizes the margin is denoted by $\delta(\mathbf{b})$. The "quantity which depends on the corrupted data" mentioned in the approximate expression (22) of the error bound is thus the square root in the numerator of (34), namely the square root of the sum of the number $n(\mathbf{b})$ of times each corrupted piece of data **b** is fed to the learner, weighted by (the square of) the distance $\delta(\mathbf{b})$ plus the distance $\mu(\mathbf{A})$.

REFERENCES

Tamás Sándor BÍRÓ (2006), *Finding the right words: Implementing Optimality Theory with Simulated Annealing*, Ph.D. thesis, University of Groningen, available as ROA-896.

Hans-Dieter BLOCK (1962), The perceptron: A model of brain functioning, *Review of Modern Physics*, 34(1):123–135.

Paul BOERSMA (1997), How we learn variation, optionality and probability, in Rob VAN SON, editor, *Proceedings of the Institute of Phonetic Sciences (IFA) 21*, pp. 43–58, Institute of Phonetic Sciences, University of Amsterdam.

Paul BOERSMA (1998), *Functional Phonology*, Ph.D. thesis, University of Amsterdam, The Netherlands, holland Academic Graphics.

Paul BOERSMA and Bruce HAYES (2001), Empirical tests for the Gradual Learning Algorithm, *Linguistic Inquiry*, 32(1):45–86.

Paul BOERSMA and Joe PATER (to appear), Convergence properties of a gradual learner for Harmonic Grammar, in John MCCARTHY and Joe PATER, editors, *Harmonic Grammar and Harmonic Serialism*, Equinox Press.

Paul BOERSMA and Jan-Willem VAN LEUSSEN (2014), Fast evaluation and learning in multi-level parallel constraint grammars, University of Amsterdam.

Nicolò CESA-BIANCHI and Gábor LUGOSI (2006), *Prediction, learning, and games*, Cambridge University Press.

Andries W. COETZEE and Shigeto KAWAHARA (2013), Frequency biases in phonological variation, *Natural Language and Linguistic Theory*, 31(1):47–89.

Andries W. COETZEE and Joe PATER (2008), Weighted constraints and gradient restrictions on place co-occurrence in Muna and Arabic, *Natural Language and Linguistic Theory*, 26(2):289–337.

Andries W. COETZEE and Joe PATER (2011), The place of variation in phonological theory, in John GOLDSMITH, Jason RIGGLE, and Alan YU, editors, *Handbook of phonological theory*, pp. 401–434, Blackwell.

Nello CRISTIANINI and John SHAWE-TAYLOR (2000), *An introduction to Support Vector Machines and other kernel-based methods*, Cambridge University Press.

Robert FRANK and Shyam KAPUR (1996), On the use of triggers in parameter setting, *Linguistic Inquiry*, 27(4):623–660.

Yoav FREUND and Robert E. SCHAPIRE (1999), Large margin classification using the Perceptron algorithm, *Machine Learning*, 37(3):277–296.

Edward GIBSON and Kenneth WEXLER (1994), Triggers, *Linguistic Inquiry*, 25(3):407–454.

Bruce HAYES (2004), Phonological acquisition in Optimality Theory: The early stages, in René KAGER, Joe PATER, and Wim ZONNEVELD, editors, *Constraints in phonological acquisition*, pp. 158–203, Cambridge University Press.

Gaja JAROSZ (2013), Learning with hidden structure in Optimality Theory and Harmonic Grammar: Beyond Robust Interpretative Parsing, *Phonology*, 30(1):27–71.

Karen JESNEY and Anne-Michelle TESSIER (2011), Biases in Harmonic Grammar: the road to restrictive learning, *Natural Language and Linguistic Theory*, 29(1):251–290.

Frank KELLER (2000), *Gradience in grammar. Experimental and computational aspects of degrees of grammaticality*, Ph.D. thesis, University of Edinburgh, England.

Jyrki Kivinen (2003), Online learning of linear classifiers, in Shahar Mendelson and Alexander J. Smola, editors, *Advanced lectures on Machine Learning (LNAI 2600)*, pp. 235–257, Springer.

Jyrki Kivinen, Manfred K. Warmuth, and Peter Auer (1997), The Perceptron algorithm versus Winnow: linear versus logarithmic mistake bounds when few input variables are relevant, *Artificial Intelligence*, 97(1–2):325–343.

Norbert Klasner and Hans-Ulrich Simon (1995), From noise-free to noise-tolerant and from on-line to batch learning, in Wolfgang Maass, editor, *Computational Learning Theory (COLT) 8*, pp. 250–257, ACM.

Gèraldine Legendre, Yoshiro Miyata, and Paul Smolensky (1998a), Harmonic Grammar: A formal multi-level connectionist theory of linguistic well-formedness: An application, in Morton Ann Gernsbacher and Sharon J. Derry, editors, *Annual conference of the Cognitive Science Society 12*, pp. 884–891, Lawrence Erlbaum Associates.

Géraldine Legendre, Yoshiro Miyata, and Paul Smolensky (1998b), Harmonic Grammar: A formal multi-level connectionist theory of linguistic well-formedness: Theoretical foundations, in Morton Ann Gernsbacher and Sharon J. Derry, editors, *Annual conference of the Cognitive Science Society 12*, pp. 388–395, Lawrence Erlbaum.

Gèraldine Legendre, Antonella Sorace, and Paul Smolensky (2006), The Optimality Theory/Harmonic Grammar connection, in Paul Smolensky and Gèraldine Legendre, editors, *The Harmonic Mind*, pp. 903–966, MIT Press.

Nick Littlestone (1988), Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm, *Machine Learning*, 2(4):285–318.

Giorgio Magri (2015), Idempotency in Optimality Theory, manuscript.

Giorgio Magri (to appear), Error-driven learning in OT and HG: a comparison, *Phonology*.

Marvin Minsky and Seymour Papert (1969), *Perceptrons: An introduction to Computational Geometry*, MIT Press.

Mehryar Mohri and Afshin Rostamizadeh (2013), Perceptron mistake bounds, arXiv:1305.0208.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar (2012), *Foundations of Machine Learning*, MIT Press.

Albert B. J. Novikoff (1962), On convergence proofs on Perceptrons, in *Proceedings of the symposium on the mathematical theory of automata*, volume XII, pp. 615–622.

Joe Pater (2008), Gradual learning and convergence, *Linguistic Inquiry*, 39(2):334–345.

Alan PRINCE (2002), Entailed Ranking Arguments, ms., Rutgers University, New Brunswick, NJ. Rutgers Optimality Archive, ROA 500. Available at http://www.roa.rutgers.edu.

Alan PRINCE and Bruce TESAR (2004), Learning phonotactic distributions, in René KAGER, Joe PATER, and Wim ZONNEVELD, editors, *Constraints in phonological acquisition*, pp. 245–291, Cambridge University Press.

Frank ROSENBLATT (1958), The Perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review*, 65(6):386–408.

Frank ROSENBLATT (1962), *Principles of Neurodynamics*, Spartan.

Shai SHALEV-SHWARTZ and Yoram SINGER (2005), A new perspective on an old Perceptron algorithm, in Peter AUER and Ron MEIR, editors, *Conference on Computational Learning Theory (COLT) 18*, Lecture notes in Computer Science, pp. 264–278, Springer.

Paul SMOLENSKY and Gèraldine LEGENDRE (2006), *The Harmonic Mind*, MIT Press.

Kenneth WEXLER and Peter W. CULICOVER (1980), *Formal principles of language acquisition*, MIT Press, Cambridge, MA.

Economy of Expression as a principle of syntax

Mary Dalrymple[1]*, Ronald M. Kaplan*[2]*, and Tracy Holloway King*[3]
[1] Faculty of Linguistics, Philology and Phonetics, University of Oxford
[2] Nuance Communications
[3] A9.com

ABSTRACT

Keywords: Lexical Functional Grammar, Economy of Expression

The purpose of a grammatical theory is to specify the mechanisms and principles that can characterize the relations of acceptable sentences in particular languages to the meanings that they express. It is sometimes proposed that the simplest and most explanatory way of arranging the formal mechanisms of grammatical description is to allow them to produce unacceptable representations or derivations for some meanings and then to appeal to a global principle of economy to control this overgeneration. Thus there is an intuition common to many syntactic theories that a given meaning should be expressed in the most economical way, that smaller representations or shorter derivations should be chosen over larger ones.

In this paper we explore the conceptual and formal issues of Economy as it has been discussed within the theory of Lexical Functional Grammar. In LFG the metric of Economy is typically formulated in terms of the size of one component of syntactic representation – the surface constituent structure tree – but it is often left unstated which trees for a given meaning are to be compared and how they are to be measured. We present a framework within which alternative explicit definitions of Economy can be formulated, and examine some phenomena for which Economy has been offered as an explanation. However, we observe that descriptive devices already available and independently motivated within the traditional LFG formalism can also account for these phenomena directly, without relying

Journal of Language Modelling Vol 3, No 2 (2015), pp. 377–412

on cross-derivational comparisons to compensate for overgeneration. This leads us to question whether Economy is necessary or even useful as a separate principle of grammatical explanation.

1 INTRODUCTION

There is an intuition common to many syntactic theories that a given meaning must be expressed in the most economical way: that only smaller representations or shorter derivations should be classified as well-formed, and larger expressions of the same meaning should be discarded. In implementing this intuition, it is sometimes proposed that the simplest and most explanatory way of arranging the formal mechanisms of grammatical description is to allow them to produce unacceptable representations or derivations for some meanings and then to appeal to a general grammatical principle to control this overgeneration. Economy classifies a derivation as grammatical if and only if it is among the smallest or most economical according to the relevant Economy metric, and non-economical expressions of the same meaning are classified as ungrammatical.

For all theories of syntax, the question arises of whether there is a global Economy principle classifying derivations as grammatical or ungrammatical. In defining Economy any theory needs to consider (1) the candidate representations that provide the choice space for Economy, and (2) the nature of the strings that are involved in Economy comparisons. Different theories may appeal to different metrics in defining Economy; for some theories, the number of steps in a derivational process may be the relevant measure, while in other theories the number of nodes in a constituent structure tree or the number of components of some other grammatical structure may be relevant. Optimality-theoretic (OT) approaches (Morimoto 2001; Grimshaw 2001) assume a general constraint on expression that identifies smaller structures as grammatical in comparison to larger ones, and Collins (2003) discusses a class of what he calls "Economy of Representation" approaches which propose similar constraints, e.g. Emonds' slogan "Use as few words as possible" (Emonds 1994).

In this paper we present a formal framework within which alternative explicit definitions of an Economy principle can be examined, cast within the theory of Lexical Functional Grammar (LFG: Kaplan and Bresnan 1982). The metric of Economy as discussed in the LFG literature is typically formulated in terms of the size of one component of syntactic representation, the surface constituent structure tree, but it is often left unstated exactly which trees for a given meaning are to be compared and precisely how they are to be measured. Our aim is to shed light on the nature and definition of Economy; in doing so, we raise some issues about the nature of Economy as a principle of grammar, and call into question the necessity of such a principle.

Economy vs. pragmatic, stylistic, or processing-based metrics

It is important to separate the Economy metric from other stylistic, pragmatic, or processing-based preferences that may also value succinctness or brevity. According to Economy, the only grammatical means of expressing a given meaning are the smallest ones, and larger ones are classified as ungrammatical and discarded. Other linguistic modules may be involved in comparing ways of expressing broadly similar meanings: for example, Gricean maxims of quantity or manner (Grice 1975) may prefer more succinct expressions of a particular meaning over less succinct ones. Similarly, comparisons among grammatical derivations may be important in language acquisition and processing (Kuhn 1999, among many others), and such considerations may provide evidence for processing-based preferences or selection of particular grammatically well-formed structures over others. However, such preferential mechanisms always choose among grammatically well-formed expressions of the relevant meaning, each of which (according to the Economy principle) is among the smallest for the particular meaning it expresses. Since pragmatic, stylistic, or processing-based preferences choose only among grammatical utterances, they are orthogonal to the Economy-based classification of utterances as grammatical or ungrammatical upon which we focus.

Economy vs. Blocking

We also distinguish Economy as a syntactic metric from Blocking (Andrews 1990; Bresnan 2003; Embick and Marantz 2008) as a morphological metric. Though both Blocking and Economy involve competition among different ways of expressing a particular meaning, the vast majority of cases of morphological blocking involve comparison between single words, for example **goed* vs. *went*. In contrast, the

Economy metric in LFG evaluates alternative constituent structure trees, choosing smaller trees and rejecting larger trees; it is not considered when making the choice between alternative single words appearing in the same position in the same syntactic structure. Economy is, however, relevant for a particular subset of morphological blocking cases: those which have been termed "Poser blocking" (Poser 1992; Embick and Marantz 2008), where the availability of a single-word expression of a particular meaning is claimed to block the expression of that meaning as a multi-word phrase; we discuss Poser blocking in Section 6.2.

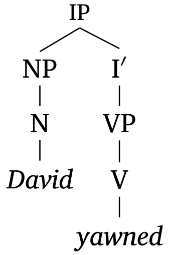
In Section 2, we introduce LFG, principle-based specification of LFG grammars, and explanatory concerns for the theory of syntax in adopting an Economy metric. We provide the background and definitions for our formal account of Economy in Section 3, proposing three alternative definitions of how Economy is measured. In the following three sections, we explore each of these three definitions, discuss how they relate to previous proposals, and evaluate some empirical evidence that has been proposed as motivation for each definition.

Based on our formalization of Economy and its proposed application to several phenomena that have been taken to motivate such a principle, we do not find Economy to be a compelling explanatory principle of grammar, at least from the perspective of LFG. Economy is unlike other commonly assumed grammatical principles in involving a global comparison among otherwise well-formed structures, rather than well-formedness conditions that must be met by grammatical structures or rules. Hence, the burden of proof is on proponents of Economy to show that its effects cannot be achieved by independently-motivated, pre-existing grammatical mechanisms. Our examination of some of the cases that have been taken to support an Economy metric reveal that alternative accounts are in fact available, and we suggest that a convincing case for Economy has not yet been made.

2 SPECIFICATION OF LFG GRAMMARS AND THE NATURE OF ECONOMY

An LFG grammar assigns to every string in its language at least one functional structure (f-structure) that corresponds to at least one constituent structure tree (c-structure). The constituent structure tree represents linear order and phrasal grouping, while the functional structure represents abstract predicate-argument relations and information about case, agreement, tense, and other grammatical features. The c-structure and simplified f-structure for *David yawned* is given in (1):

(1) Constituent structure: Functional structure:



$$\left[\begin{array}{ll} \text{PRED} & \text{‘YAWN}\langle\text{SUBJ}\rangle\text{’} \\ \text{TENSE} & \text{PAST} \\ \text{SUBJ} & \left[\begin{array}{ll} \text{PRED} & \text{‘DAVID’} \\ \text{PERS} & 3 \\ \text{NUM} & \text{SG} \end{array}\right] \end{array}\right]$$

An f-structure f belongs to a set F of attribute-value matrices that satisfy all of the f-structure well-formedness conditions specified by LFG theory, including at least the Uniqueness, Coherence, and Completeness conditions.[1] Similarly, a c-structure c belongs to a set C of valid phrase structure trees that satisfy additional well-formedness conditions: traditionally these include the formal prohibition against non-branching dominance chains (Kaplan and Bresnan 1982), though additional constraints, such as those requiring X′-theoretic configurations or the disallowance of empty nodes, have also been explored, as we discuss below. However the well-formedness conditions might be specified, the elements of F and C are the "valid" structures with respect to LFG theory: they are the only ones that serve as models of grammatical constraints and thus the only ones that figure in a meaningful discussion of grammar-based Economy.

An LFG grammar is traditionally specified by a system of node-admissibility constraints presented in the rewriting-rule format of a context-free grammar (Dalrymple *et al.* 1995a). The daughters in each rule are decorated with functional schemata, and these are instantiated to constraints on the corresponding f-structures. The f-structures are valid models for the functional constraints that are associated with at least one c-structure.

[1] The Uniqueness condition guarantees that each attribute in an f-structure has exactly one value. Completeness and Coherence guarantee that the valency requirements of each syntactic predicate are satisfied appropriately.

An LFG grammar can be specified in other ways, however. It can be specified by a collection of more abstract conditions or "principles" that the grammar must satisfy. These grammatical principles are different from the well-formedness conditions on c-structures (such as the Non-Branching Dominance constraint) and f-structures (Completeness, Coherence, and Uniqueness) that all LFG grammars assume. Rather, such principles characterize the properties that grammar rules and lexical entries must have in order to be admissible in a well-formed grammar. For example, Bresnan (2001) proposes endocentricity principles to characterize possible arrangements of categories in c-structure rules, and structure-function mapping principles to indicate how functional schemata are distributed onto the c-structure rules. According to one such principle (Bresnan 2001, 103), a projecting node in a projection of the same kind (that is, a head) is annotated with $\uparrow = \downarrow$, meaning that a phrase and its head must correspond to the same f-structure. On this view, any traditional rule that satisfies the principles is assumed to be a well-formed rule of grammar, and rules that do not obey these principles are disallowed.

To be precise, for a grammar specified by means of a collection of grammatical principles $\mathscr{G}$ to be interpretable within an LFG framework, there must be a traditional grammar $G_{\mathscr{G}}$ that consistently realizes all of $\mathscr{G}$'s stipulations. We can then investigate the impact of alternative Economy proposals by examining the corresponding traditional LFG grammars $G_{\mathscr{G}}$ in which annotated c-structure rules and lexical entries are enumerated explicitly. For instance, Toivonen's principles of phrase structure differ from Bresnan's in requiring a strict version of X′ theory, without allowing for X′ elision as described below. The details of the concrete LFG grammars are the basis for evaluating and comparing different Economy proposals.

2.1 *Economy and the optionality provision*

The Economy proposals of both Bresnan (2001) and Toivonen (2003) include a general provision that nodes that are obligatory according to other rules and principles are omitted from c-structure if semantic expressiveness and certain other syntactic conditions can be maintained without them. We can formalize two special cases of the optionality provision: the systematic omission of daughter nodes and the elision of nonbranching X′ nodes.

2.1.1 Daughter omission

The convention of Daughter Omission stipulates that all daughters in a c-structure rule are optional:

(2) Daughter Omission:

If an LFG grammar $G_{\mathscr{G}}$ contains an annotated rule of the form

$$\text{Y} \rightarrow \alpha\ \text{Z}\ \beta$$

(where α or β may be the empty string ε), it also contains a rule of the form

$$\text{Y} \rightarrow \alpha\ \beta$$

Thus, if the grammar (or a set of abstract grammatical principles) sanctions a rule such as (3a), independently omitting each of the daughters would provide for the additional rules (3b-d) and for the smaller trees that they would allow. These could be expressed in a single rule by using the parentheses notation that indicates optionality in traditional LFG grammars, as in (3e).

(3)

$$\begin{array}{lllll}
\text{a.} & \text{V}' & \longrightarrow & \text{V} & \text{NP} \\
 & & & \uparrow=\downarrow & (\uparrow \text{OBJ})=\downarrow \\
\text{b.} & \text{V}' & \longrightarrow & \text{V} & \\
 & & & \uparrow=\downarrow & \\
\text{c.} & \text{V}' & \longrightarrow & \text{NP} & \\
 & & & (\uparrow \text{OBJ})=\downarrow & \\
\text{d.} & \text{V}' & \longrightarrow & \varepsilon & \\
\text{e.} & \text{V}' & \longrightarrow & (\ \text{V}\ ) & (\ \text{NP}\ ) \\
 & & & \uparrow=\downarrow & (\uparrow \text{OBJ})=\downarrow
\end{array}$$

Daughter omission in particular allows for rules that dominate no lexical material, as illustrated by (3d); we return to this point in Section 4.2.

Daughter Omission is not a necessary component or corollary of Economy: an Economy metric can be used to choose among larger and smaller derivations even when, contrary to a completely general principle of Daughter Omission, some nodes are obligatory in some configurations. Nevertheless, many researchers have adopted Daughter Omission as a central grammatical principle and see it as a key component of Economy.

2.1.2 X′ elision

Bresnan's (2001) specification of Economy allows for the omission of nodes in a broader range of configurations. Many versions of X′ theory admit nonbranching single-bar-level X′ categories whose annotations impose no constraints on the form of the corresponding f-structures. These nonbranching nodes may be optionally elided, creating alternative XP structures which do not contain an X′ node. Doing this increases the number of candidate c-structures while still permitting the same meanings to be expressed. Other things being equal, Economy selects the trees without those nodes.

(4) X′ elision:
If an LFG grammar $G_{\mathcal{G}}$ contains an annotated rule of the form

$$\begin{array}{lll} \mathrm{XP} \rightarrow \alpha & \mathrm{X}' & \beta \\ & \uparrow = \downarrow & \end{array}$$

it also contains a rule of the form

$$\begin{array}{lll} \mathrm{XP} \rightarrow \alpha & \mathrm{X} & \beta \\ & \uparrow = \downarrow & \end{array}$$

The elided X′ nodes are redundant in the sense that their appearance has no impact on either the strings of the language characterized by the grammar or their corresponding f-structures. X′ elision is consistent with Bresnan's pretheoretic intuition that redundant c-structure nodes need not appear in grammatically well-formed c-structures and should be ruled out by Economy considerations.

Bresnan (2001, 115) observes that the redundancy intuition does not apply to all nonbranching category configurations. In particular, VP nodes under S are retained even when they are nonbranching and even though they carry the $\uparrow = \downarrow$ annotation which appears on functional heads. Bresnan's rationale for this is that there is no separate principle of structure-function mapping that would allow for the $\uparrow = \downarrow$ annotation on a V or V′ directly under S. Other principles may require omission or elision of otherwise mandatory nodes in other circumstances.

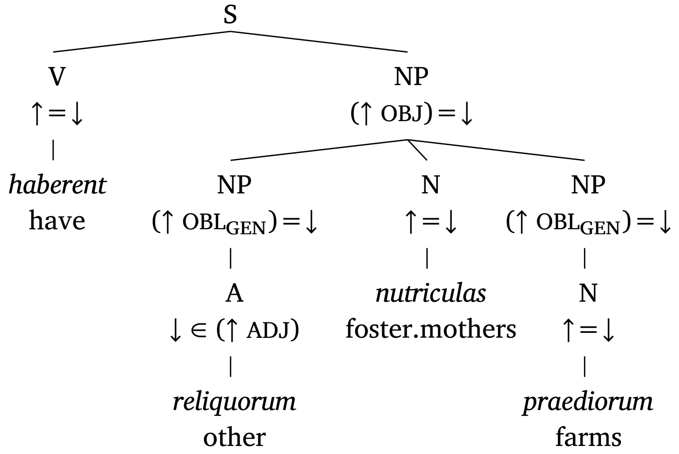
Not all LFG researchers adopt X′-elision, however. While also advocating for an Economy principle, Toivonen (2003) proposes a stricter version of the optionality provision that allows for daughters but not nonbranching X′ nodes to be omitted. Compared to Bresnan's theory, Toivonen includes fewer c-structures as candidates to be evaluated by an Economy comparison.

2.2 *Optionality and discontinuity*

Nordlinger and Sadler (2007) point out that Daughter Omission allows a simple analysis of discontinuous constituents in some languages. If the head is optional in the c-structure expansion of a category, a phrase can occur in one position without its head and in another position with its head. This is a welcome result for languages that allow discontinuity, as Snijders (2012) shows for the following Latin example (which we have adapted from Snijders's tree):

(5) a. ... haberent reliquorum nutriculas
have.3PL.IMPF.CONJ other.GEN.PL foster-mothers.ACC
praediorum.
farms.GEN
'...they might have foster mothers for their other farms.'
(Cic. Phil. 11.12, from Bolkestein 2001, 253 via Snijders 2012)

b.



Here the genitive oblique 'other farms' does not form a constituent; the adjective *reliquorum* 'other' is separated from the noun it modifies by the noun *nutriculas* 'foster mothers'. Since the head noun N is optional in the NP subtree, the adjective can appear as an NP constituent on its own, with the head N in a separate NP. Since the two NP nodes have the same annotation $(\uparrow \textsc{obl}_{\textsc{gen}}) = \downarrow$, they contribute to the same f-structure.

John Lowe (p.c.) further observes that headless phrases can lead to spurious ambiguity in the case where "discontinous" constituents happen to be adjacent in the string. A multi-word constituent in such a situation might also be analyzed as separate but adjacent components of a single functional unit. This is shown abstractly in (6).

(6) a. Single constituent: [$_{\text{NP}}$ A N] V
b. Two adjacent constituents: [$_{\text{NP}}$ A] [$_{\text{NP}}$ N] V

We return to this point in Section 4.1, in our discussion of Same-String Economy.

Not all languages allow discontinuity, however, and additional principles must be introduced to control the appearance and distribution of headless constituents within and across languages if a fully general principle of Daughter Omission is adopted. We briefly explore some of the relevant issues in the rest of this section.

2.2.1 Free word order without discontinuity

Japanese is a free word order language, allowing the arguments of a verb to appear in any order (subject to pragmatic constraints: Fry and Kaufmann 1998). Any order of the three arguments of the verb *ageta* 'gave' is acceptable, including the two orders presented in (7):

(7) a. [Taroo ga] [yubiwa o] [kono onnanoko ni] ageta.
Taroo NOM ring ACC this girl DAT gave
'Taroo gave a ring to this girl.'

b. [kono onnanoko ni] [Taroo ga] [yubiwa o] ageta.
this girl DAT Taroo NOM ring ACC gave
'Taroo gave a ring to this girl.'

Under Daughter Omission, the head of the Japanese noun phrase is optional, as in Latin. The expectation is, then, that it should be possible to have part of the dative-marked argument *kono onnanoko ni* 'to this girl' in sentence-initial position, and part of it before the verb, since, as (7) shows, the entire phrase can appear in either position. However, this is not possible: splitting the noun phrase into two parts is unacceptable, whether or not the dative casemarker is repeated, and independent of the relative order of the two parts of the phrase. In example (8a), the noun *onnanoko* 'girl' appears sentence-initially and the determiner *kono* ‘this’ appears preverbally, and in (8b) the order is reversed; both are unacceptable.

(8) a.* [onnanoko (ni)] [Taroo ga] [yubiwa o] [kono (ni)]
girl DAT Taroo NOM ring ACC this DAT
ageta.
gave
‘Taroo gave a ring to this girl.’

b.* [kono (ni)] [Taroo ga] [yubiwa o] [onnanoko (ni)]
this DAT Taroo NOM ring ACC girl DAT
ageta.
gave
‘Taroo gave a ring to this girl.’

In contrast, if we do not assume a completely general version of Daughter Omission, this problem is avoided by assuming that the difference between Latin and Japanese is that phrasal heads are optional in Latin, but obligatory in Japanese. If a noun phrase cannot appear without its noun head, discontinuity is disallowed and the examples in (8) are correctly ruled out.

Joan Bresnan (p.c.) raises the possibility that the crucial difference between Latin and Japanese lies not in head obligatoriness, but in principles for rule annotation in each language. In Latin, more than one phrase in a single clause can be annotated with the same grammatical function, while in Japanese only one nominal phrase per clause may be annotated with any particular grammatical function. For example (8), treating the difference between Latin and Japanese in terms of differences in permitted annotations on the daughter nodes of clausal categories would successfully control the availability of discontinuous phrases where both components of the phrase are daughters of the same clausal category.[2]

However, when taking this view, it is not clear how adjuncts can be treated, nor how one might predict adjunct discontinuity. In standard LFG treatments, the annotation $\downarrow \in (\uparrow \text{ADJ})$ appears on all adjuncts, indicating that the f-structure for the phrase bearing the annotation should appear in the set of adjuncts of the f-structure of the

[2] More complex constraints would be required to forbid discontinuity involving nonsister components, but this may also be possible through the use of special phrase structure categories or additional annotations on rules.

mother node. Restricting this annotation to appear only once incorrectly predicts that only one adjunct can appear. On the other hand, allowing this annotation to appear more than once, while maintaining Daughter Omission, predicts that adjuncts, and only adjuncts, can be discontinuous in languages like Japanese. Neither prediction is correct, and it is not clear how the proposal can be modified to allow for the correct treatment of both arguments and adjuncts.

Of more significance, however, is the theoretical difficulty of this proposal: it reduces the generality of the annotation principles and weakens their explanatory power. It admits the possibility that annotations can be parametrized to allow or disallow discontinuity or other variations in language-particular or construction-specific ways.

2.2.2 Obligatoriness even where discontinuity is otherwise allowed

A further problem for Daughter Omission is raised by Snijders (2012), who provides an analysis of Latin phrase structure and proposes that the correct analysis must treat some nodes as obligatory. Following Bolkestein (2001), Snijders (2012) shows that the following constraint holds in Latin:

(9) Constraint on Latin discontinuous NPs:
No discontinuity is allowed between a P and the NP it governs (yet the NP may be internally discontinuous, meaning that part of the NP may be separated from the P).

Example (5) establishes that an NP constituent in Latin need not contain an N: this correctly allows for discontinuous nominal phrases, under the assumption that Bresnan's proposed analysis of Japanese, where an annotation for a particular grammatical role can appear only once, does not apply to Latin. However, Snijders shows that the generalization in (9) must be analyzed by specifying the NP complement within a PP as obligatory: some portion of the NP complement, not necessarily including the head, must appear adjacent to the P. If the NP complement of PP were optional, the P would be able to appear on its own, not adjacent to any component of its complement.[3] In sum, though optionality is well-attested in many constructions and in many languages, Daughter Omission appears to be non-viable as a general, exceptionless principle.

[3] See Snijders (2012) for further discussion and exemplification.

In the current context, our key point is that the adoption of particular grammatical principles such as Daughter Omission is orthogonal to the adoption of an Economy metric. That is, adopting a principle of Daughter Omission does not require the concomitant adoption of Economy to choose among larger and smaller candidate trees. Conversely, adoption of an Economy principle is compatible with a theoretical view which rejects Daughter Omission and allows obligatory phrase structure nodes. The purpose of an Economy metric is to select derivations with smaller and therefore more desirable c-structures from among all of the derivations that a grammar (with or without Daughter Omission or other optionality principles) produces.

2.3 *The Economy principle as a cross-derivational constraint*

Economy as a principle of grammar has a different status from other grammatical principles and conditions. Economy is not a well-formedness condition on individual c-structures or f-structures (like Completeness or Coherence), nor is it a constraint on the form of possible grammar rules (like Bresnan's structure-function mapping principles). Instead, it is a global, cross-derivational constraint, classifying structures as ungrammatical that may be well-formed according to the other grammatical principles and conditions, but which are not the smallest such structures to express a particular meaning. This stands in sharp contrast to the LFG convention of assigning to a sentence the minimal f-structure satisfying its functional description or to the substantially equivalent provision of Construction Grammar that only fully-licensed representations are admissible (Kay 2002). The minimal f-structure can be determined by the incremental evaluation of the constraints of a single derivation's f-description without reference to the descriptions or structures of other derivations.[4]

[4] It is also important to recognize that selecting the minimal f-structure for a particular LFG derivation is essentially unrelated to the notion of Economy of Expression. As we will point out in Section 3.3, an f-structure corresponding to a specific meaning forms the basis for the Economy comparison, and the issue is which of any competing strings are assigned to that f-structure by the derivation relation Δ_G. The given f-structure may not be minimal with respect to the derivations of some of those strings, in which case those derivations fail on their own merits without comparison to other strings or derivations. They are simply disallowed as ways of expressing the meaning encoded in the given f-structure.

Potts (2002) points out that the machinery of cross-derivational comparisons substantially increases the logical complexity of several linguistic theories, including LFG. It requires a mathematical layer on top of the standard formal devices, mechanisms and other principles of grammar, and therefore introduces a significant — and not well understood — expansion of the expressive power of grammatical description. For this reason it is not something to be taken on without very careful justification. And at least with respect to other theories, Potts cites a range of papers that call into question its empirical consequences.

Economy may serve as an informal but useful summary for a collection of grammatical relationships without actually being posited as an independent operational linguistic principle. That is, it is perhaps best interpreted as a generalization about the combined effect of other principles and grammatical mechanisms, each functionally and/or psycholinguistically motivated, that together give rise to the appearance of a very general principle favoring smaller structures over larger structures. On this view, Economy is not an independent constraint but a by-product of formal devices and principles that must already be deployed in grammars of individual languages.

It is not clear whether Economy is a necessary or sufficient principle of grammar, and just its logical complexity militates against its inclusion in the theory of syntax. Thus, with Potts (2002), we suggest that the burden of proof is on proponents of Economy to show that such a fully general principle of comparison is not merely an illusion stemming from the operation of separately motivated mechanisms and principles that must be assumed in any case.

3 ECONOMY AND THE FORMAL STRUCTURE OF LINGUISTIC DERIVATIONS

Any theory in which an Economy principle plays a role must make explicit the structures that are candidates for the Economy comparison and how such structures are selected. In this section we offer the definitions necessary for a formal account of Economy in an LFG setting.

3.1 *LFG grammars as constraints over grammatical structures*

Wedekind and Kaplan (2012) observe that an LFG grammar G characterizes a derivation relation Δ_G over string/f-structure pairs. They offer essentially the following definition:

(10) The derivation relation Δ_G

$\Delta_G(s, f)$ iff G assigns to the string s the f-structure $f \in F$, where F is the set of well-formed f-structures.

We extend this definition so that Δ_G explicitly takes account of the c-structure:

(11) The derivation relation Δ_G (extended)

$\Delta_G(s, c, f)$ iff G assigns to the string s the c-structure $c \in C$ and f-structure $f \in F$.

3.2 *The generation set for a grammar G*

All definitions of Economy involve a comparison among alternative means of expressing a common meaning m. We define $Exp(m)$ as the set of f-structures that express a meaning m:

(12) F-structures that express a meaning m

$Exp(m) = \{f \in F \mid f \text{ expresses } m\}$

We make no assumptions here about the nature of meaning representations (logical formulas, attribute-value matrices, or other formal structures). We require only that all of the f-structures in $Exp(m)$ express the target meaning m.

C-structure and f-structure are not the only linguistic levels assumed in many LFG-based proposals: rather, a variety of linguistic properties are spread out among a collection of related structures (e.g. information structure, discourse structure, prosodic structure: Kaplan 1987; Asudeh 2006; Dalrymple and Mycock 2011; Mycock and Lowe 2013) in addition to the syntactic predicate-argument dependencies that are typically represented in f-structure. For simplicity, in this paper we consider the f-structure as standing for all grammatical information that is relevant for the Economy ranking and not represented by c-structure.

Given the definition of the meaning-expression set $Exp(m)$ in (12), the overt expression of a target meaning m is formalized as the $\langle s, c, f \rangle$ triples that the grammar G assigns to any of the f-structures in $Exp(m)$. Again extending a definition of Wedekind and Kaplan (2012), this can be formalized as the generation set $Gen_G(m)$:

(13) The generation set $Gen_G(m)$ for a target meaning m, given a grammar G

$$Gen_G(m) = \{\langle s, c, f \rangle | f \in Exp(m) \text{ and } \langle s, c, f \rangle \in \Delta_G\}$$

This is specified for a grammar G in traditional LFG notation, but as indicated above, that grammar may be a standard grammar $G_{\mathscr{G}}$ interpreting a more abstract grammatical specification $\mathscr{G}$. The generation set for $\mathscr{G}$ is defined in the obvious way:

(14) $Gen_{\mathscr{G}}(m) = Gen_{G_{\mathscr{G}}}(m)$

That is, the generation set for a target m given an abstract grammar specification $\mathscr{G}$ is the generation set for m given the traditional grammar $G_{\mathscr{G}}$ that properly interprets the abstract one.

3.3 The Economy ordering on $Gen_G(m)$

Economy compares members of the generation set for a meaning m, under the assumption that a grammar (especially one presented abstractly) may include structures containing superfluous or unwanted elements. The intended effect of Economy is to identify a smaller generation set that contains only the linguistically motivated structures. This is formalized in terms of an Economy ordering $\leq$ on $Gen_G(m)$:

(15) The Economy ordering

$$\langle s, c, f \rangle \leq \langle s', c', f' \rangle \text{ iff } \langle s, c, f \rangle \text{ is more economical than } \langle s', c', f' \rangle$$

Alternative ways of defining the Economy ordering impose different constraints on the strings s and s' but all involve comparing the sizes of the c-structures c and c'. As for the f-structures f and f', we argue below that they must be identical. Thus, the general form of the metric is given in (16), where $c \leq_c c'$ if and only if the number of relevant nodes in c is less than or equal to the number of relevant nodes in c'.[5]

[5] Proponents of Economy do not generally agree on which nodes are relevant to defining the Economy ordering. According to Bresnan (2001, 91), for example, terminal and preterminal nodes are ignored. We return to this issue in Section 6.3.

(16) General schema for the Economy ordering $\leq$

$\langle s, c, f \rangle \leq \langle s', c', f' \rangle$ iff $c \leq_c c'$ and *String_rel*(s, s') and $f = f'$

We represent constraints on the strings s, s' by the two-place relation *String_rel*. In Section 3.5 we consider a set of alternative definitions of *String_rel* that lead to different theoretical and descriptive consequences.

The $f = f'$ condition addresses the fact that the set $Exp(m)$ may contain distinct f-structures corresponding to ways of expressing a meaning m that should stand in free variation with respect to an Economy comparison. It would be descriptively incorrect, for example, if passive realizations of a given meaning were systematically suppressed in favor of their putatively more economical active counterparts. As another example, an unrestricted version of Economy might suppress the longer prepositional realization for verbs such as *give* (*He gave the book to her*) in favor of the equally acceptable but shorter ditransitive realization (*He gave her the book*). These unintended consequences could be avoided, of course, by postulating (perhaps subtle) differences in meanings that otherwise share the same underlying predicate-argument specifications. Because our formalization distinguishes meanings from the f-structures that express them, it allows alternative realizations for the same meaning to be derived from f-structures with distinct syntactic (e.g. active vs. passive) features. Restricting the domain of the Economy ordering to triples with identical f-structures thus provides for a natural account of free syntactic variation. This is consistent with the proposal of Toivonen (2003, 199) that "Economy only holds over c-structures with identical f-structure".

Bresnan (2001, 91) extends the number of derivation triples under consideration by appealing to a subsumption relation between f-structures in her definition of Economy, proposing that "a phrase structure node is omitted if the f-structure arising in its absence is at least as specific as the f-structure arising in its presence"; that is, Bresnan's definition requires that $f' \sqsubseteq f$. We note that in the special case that the smaller tree c is a subtree of the larger tree c' (and there are no disjunctive annotations on the nodes of the two trees), the monotonic mapping between c-structures and f-structures implies that $f \sqsubseteq f'$, and thus that the two f-structures are identical (since mutually subsuming f-structures are identical). Bresnan (2001) does not specifically motivate this condition on the two structures under comparison, and in the particular cases she discusses, the f-structures for the smaller and larger trees are identical and not in an asymmetric subsumption relation. Thus we see no argument against the simpler and more restrictive requirement that $f = f'$.

3.4 *Economical elements of* $Gen_G(m)$

Once we have established the Economy ordering, we can identify certain $\langle s, c, f \rangle$ triples as the minimal, most economical elements of $Gen_G(m)$, given a grammar G and a target structure m:

(17) Minimal elements of $Gen_G(m)$

A triple $\langle s, c, f \rangle$ is a *minimal* element of $Gen_G(m)$ iff no $Gen_G(m)$ element is smaller according to the Economy ordering relation $\leq$.

Economy classifies the minimal elements of $Gen_G(m)$ as grammatical, and the nonminimal elements in $Gen_G(m)$ as ungrammatical.

3.5 *Variant definitions of Economy*

We now have a formal framework for characterizing and comparing the alternative notions of Economy: which structures are in the domain of the Economy ordering $\leq$, and precisely how that ordering is defined on the elements within its domain. We provide the following three alternative definitions of *String_rel*, differing as to whether (1) all alternative c-structures for the same string are compared, (2) all alternative c-structures with the same set of terminal nodes are compared, or (3) c-structures over strings with possibly different terminals are compared.

Same-String Economy compares different c-structures over the same string.

(18) Same-String Economy ordering

String_rel(s, s') iff $s = s'$

Each string that expresses the target meaning is associated by Same-String Economy with the smallest c-structure that analyzes it, but there is no Economy comparison between c-structures for different strings.

String-Permutation Economy compares c-structures with the same terminal nodes, but possibly in a different order.

(19) String-Permutation Economy ordering[6]

String_rel(s,s') iff $s \in Perm(s')$

String-Permutation Economy allows comparison of c-structures over permutations of the same string. There is no Economy comparison between c-structures over strings that are not related by permutation.

Different-Words Economy compares c-structures without placing any restriction on the strings that each c-structure analyzes. The smallest c-structures that express the target meaning are chosen by Different-Words Economy, and strings are ruled out that express the target meaning but are not analyzed by economical trees. In this case the relation *String_rel*(s,s') holds vacuously for any pair of strings s and s'.

Relations among the definitions

There is an implicational relation among these three definitions, since the comparison is over increasingly larger sets of c-structures corresponding to the same target meaning. Given these implicational relations, any comparison that is relevant for Same-String Economy is also relevant for String-Permutation Economy and Different-Words Economy, and similarly any comparison that is relevant for String-Permutation Economy is also relevant for Different-Words Economy.

In the following sections, we explore each of these three definitions and their consequences. We show that several previously proposed definitions of Economy instantiate different definitions of the string requirement *String_rel* while still adhering to the general definition of Economy as given in (16).

4 SAME-STRING ECONOMY: SPURIOUS AMBIGUITY AND EMPTY CATEGORIES

4.1 *Same-String Economy and spurious ambiguity*

Certain sets of $Gen_G(m)$ triples for a grammar G differ only in c-structure, and have exactly the same string and f-structure.[7] These represent c-structure ambiguities that do not correlate with differences at

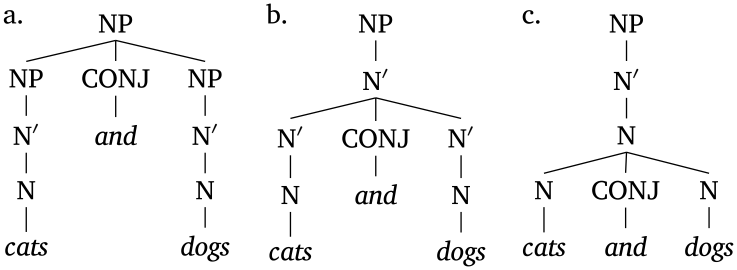
[6] $s \in Perm(s')$ iff s is a permutation of s'.

[7] Recall from Section 3.2 that we consider the f-structure to stand for all relevant levels of linguistic structure other than c-structure.

any other level of structure, since in such cases the choice of a particular c-structure has no effect on the relation established by the grammar between strings and f-structures. Such ambiguities are ruled out by all versions of Economy.

We mentioned in Section 2.2.1 that spurious ambiguities can arise over the same string if the elements of a putatively discontinuous functional unit appear next to each other in the string (John Lowe, p.c.). Spurious ambiguity also commonly occurs with single-word coordinated phrases. If coordination is possible at any X′ level, all three trees in (20) are possible:

(20)



The Same-String Economy metric selects tree (20c) as the most economical, since it has fewer nodes than tree (20a) or (20b). As argued by Frank (2006), Economy of Expression would also prefer a symmetric coordination analysis for German VP coordination over an asymmetric analysis when both are possible, because the asymmetric structure contains more nodes than the symmetric structure, and both structures correspond to the same f-structure.

There is an alternative way of viewing classes of derivations that differ only in c-structure and cannot be empirically distinguished in any other way. Rather than relying on a principle like Economy to choose the *smallest* member of a set of derivations that are indistinguishable except for the size of the c-structure, we can recognize that the alternatives arise only as an artifact of our internal derivational machinery. On this view there is no theoretical or empirical reason to prefer one candidate over another, and we can thus dispense with the need to make a choice between such otherwise equivalent derivations. We formalize an equivalence relation on derivations in the obvious way, by abstracting over c-structure variation:

(21) Equivalence relation on derivations

For all $d = \langle s, c, f \rangle$ and $d' = \langle s', c', f' \rangle$ in Δ_G,
$d \cong d'$ iff $s = s'$ and $f = f'$

This relation induces a collection of equivalence classes over the derivations in $Gen_G(m)$, and we suggest that it is only the existence of the classes, not the individual derivations, that matter for the determination of grammaticality and ambiguity. We can present a class by listing its members (if it is finite), but it suffices to display one member of the class as its representative element. In that case one may select the smallest (most economical) element for rhetorical purposes, but in fact another less economical element may be the single most natural result of alternative computational implementations, either for parsing or generation, or for psycholinguistic or processing reasons. On this view, there is no conceptual purpose in invoking Same-String Economy considerations to choose between such equivalent derivations.

4.2 *Same-String Economy and empty categories*

Daughter Omission (Example 2, repeated here in (22)) is a key feature of Economy for both Bresnan and Toivonen: every daughter category in every c-structure rule may or may not be present in the admitted trees.

(22) Daughter Omission:

If $G_{\mathscr{G}}$ contains an annotated rule of the form

Y → α Z β

(where α and β may be empty), $G_{\mathscr{G}}$ also contains a rule of the form

Y → α β

Daughter omission allows for empty categories: rules that dominate no lexical material. Such empty nodes were used in the earliest analysis of long-distance dependencies in LFG (Kaplan and Bresnan 1982), and Bresnan (2001) still appeals to empty nodes as a way of assigning proper grammatical functions in these constructions. Since a string can contain an unbounded number of unpronounced empty categories, Economy has been proposed to ensure that empty categories are not proliferated beyond necessity and can only appear when they are required to express a given meaning. This has been one of the stronger motivations in support of Economy of Expression.

However, Kaplan and Zaenen (1989) proposed another way of making the proper assignments of grammatical functions in long-distance constructions. They establish the proper grammatical relations in terms only of f-structure constraints that characterize *functional uncertainties*. Kaplan and Zaenen's account does not rely on empty c-structure categories in particular linear positions, and in fact their analysis specifically excludes trees with empty categories from the set of valid c-structures. This view aligns itself with the large body of literature arguing against the existence of traces or empty categories (Sag and Fodor 1994; Sag 2000; Dalrymple and King 2013). Weak crossover (Postal 1971; Wasow 1979) has been a recalcitrant challenge to proponents of eliminating traces from the c-structure tree, and Bresnan (2001) points to weak crossover phenomena as the primary source of evidence for traces. However, alternative accounts of weak crossover can be based on other f-structure or c-structure properties rather than the linear position of empty categories (Dalrymple *et al.* 2007; Nadathur 2013). If vacuous category expansions as in (3d) are not needed in the analysis of long-distance dependencies, including weak crossover, and are not permitted in valid c-structures, there is no need for a principle of Economy to impose an ordering over c-structures containing empty categories.

5 STRING-PERMUTATION ECONOMY: PROJECTING X′ STRUCTURE

Toivonen (2003) proposes the following definition of Economy:

(23) Economy of Expression (Toivonen): All syntactic phrase structure nodes are optional and are not used unless required by X′-constraints or Completeness. (Toivonen 2003, 200)

In fact, restricting the Economy comparison to syntactically valid triples $\langle s, c, f \rangle$ obviates the need for concern about whether well-formedness criteria such as Completeness should be included in the definition of Economy: only valid c-structures and f-structures are considered in economy-based comparisons, and so it is not necessary to restate these conditions in defining Economy conditions. Similarly, the restriction on X′ structure is part of the definition of a well-formed c-structure in Toivonen's version of LFG; hence, it is not a distinguishing characteristic and is thus also unnecessary for the definition of Economy of Expression.

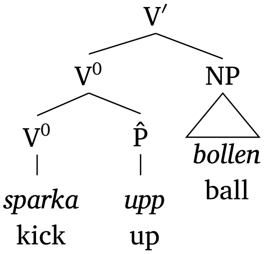
Toivonen (2003, 199) stipulates that “Economy only holds over c-structures with identical f-structure, semantic interpretation, and lexical forms”. The equivalence of semantic interpretation is already enforced by the expressivity condition of $Gen_G(m)$. Because of Toivonen’s restriction to identical words (lexical forms) in the string, her definition amounts to String-Permutation Economy:

(24) Toivonen’s Economy: String-Permutation Economy.

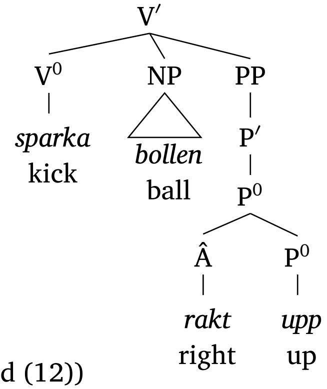
As we will see, Toivonen’s appeal to String-Permutation Economy means that her approach, unlike Poser Blocking and Bresnan’s definition of Economy (to be discussed in Section 6), does not privilege expression of meanings by words over phrases. The result is that Toivonen’s Economy comparison is defined for a smaller number of derivation triples than Poser Blocking or Bresnan’s Economy comparison.

String-Permutation Economy plays a central role in Toivonen’s (2002; 2003) analysis of word order in the Swedish VP. Toivonen proposes that prepositions and adverbs in Swedish vary as to whether they project phrasal structure. Projecting prepositions (represented as P^0) can appear after the object phrase, while nonprojecting prepositions (represented as $\hat{P}$) must adjoin to V^0. Some prepositions, such as *upp* ‘up’, are underspecified (represented simply as P), and may be either projecting or nonprojecting. For example, (25a) contains the non-projecting version of *upp*. Modifiers can only adjoin to projecting categories, so the presence of the modifier *rakt* in (25b) requires the projecting version of *upp*:

(25) a.



b.

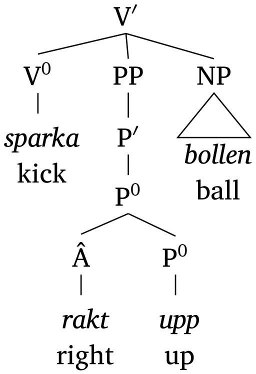


(Toivonen 2002, examples (11) and (12))

The phrase structure rules for the Swedish V′ do not allow the order V PP NP, with the result that modified particles cannot appear adjacent to the verb preceding the object:

(26) **sparka rakt upp bollen* (cf. Toivonen 2003, 101–103)

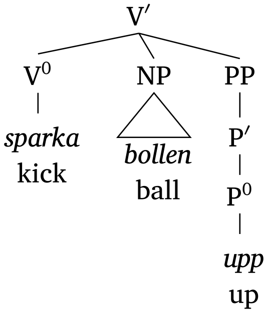
Not licensed by Swedish phrase structure rules:



Conversely, unmodified Swedish non-projecting or optionally projecting particles (unlike their English counterparts) must appear adjacent to the verb and cannot be separated from it.

(27) **sparka bollen upp* (cf. Toivonen 2003, 34–35)

Ruled out in favor of (25a) by Economy under Toivonen's account:



The ungrammaticality of example (27) cannot be explained by appeal to the V′ phrase structure rule, which allows the order V NP PP, as seen in example (25b). Furthermore, non-branching PP structures as in (27) are independently justified in Swedish in preposition stranding constructions such as interrogatives (Ida Toivonen, p.c.); these object-taking prepositions are unambiguously lexically specified as projecting and hence must appear as the P^0 head of a PP:

(28) Vem gav du boken åt?
who gave you book to
'Who did you give the book to?'

According to Toivonen's theory, String-Permutation Economy is crucial in selecting the non-projecting c-structure (25a) and ruling out the projecting structure (27).

However, there is an alternative analysis of this particular pattern which does not appeal to a global comparison under the Economy ordering. On Toivonen's analysis, lexical specifications determine whether a word is assigned the projecting category P^0 or the non-projecting category $\hat{P}$. Given the phrase structure rules of Swedish, words with the projecting category can only appear as the head of a full phrase, as in examples (25b) and (28), and non-projecting words can only appear adjoined to another head, as in example (25a). Some words, such as *upp*, are lexically ambiguous and so may appear in either position. However, when just those ambiguous words are assigned the projecting category and appear as the daughter of P', they further require the presence of a modifier. This additional requirement can be captured in standard LFG theory by annotating the P^0 categories of ambiguous words with an existential constraint $(\uparrow$ GF$)$ to guarantee the presence of a grammatical relation in the corresponding f-structure. This can be an object in the case of preposition stranding or a modifier in the case of the intransitive prepositions as in example (25b).[8] Under this alternative analysis no reference to Economy is required but the underlying intuition behind projecting and nonprojecting prepositions proposed by Toivonen is maintained.

[8] Potts (2002) also provides an alternative analysis to the Swedish data, namely that a projecting P^0 must appear in a branching PP. However, his analysis would have to be modified to account for examples with stranded prepositions, such as (28).

6 DIFFERENT-WORDS ECONOMY: AVOIDING REDUNDANT STRUCTURE

6.1 *Bresnan's Economy*

Economy of Expression is one of the major principles in Bresnan's (2001) abstract and principle-based characterization of an LFG grammar. Her principle is stated in the following way:

(29) Economy of Expression (Bresnan): All syntactic phrase structure nodes are optional and are not used unless required by independent principles (Completeness, Coherence, Semantic expressivity). (Bresnan 2001, 91)

As noted above, all definitions of Economy consider only $\langle s, c, f \rangle$ triples in which the c-structure c and f-structure f are well-formed. This observation allows us to simplify Bresnan's definition: the Completeness and Coherence conditions in Bresnan's definition are subsumed by the restriction to grammar-relevant structures. Bresnan does not provide an explicit definition of Semantic expressivity, but we understand this condition as restricting application of Economy-based comparison to the triples expressing a target meaning m, as in definition (13).

We also understand Bresnan's definition (29) together with her principles of endocentricity, structure-function mapping, etc., as specifying a traditional LFG grammar $G_{\mathscr{G}}$. The optionality provision of the Economy principle deals with the problem that the c-structure component of a $G_{\mathscr{G}}$ that realizes just the other abstract principles may not admit all trees that are linguistically desirable or necessary to express all meanings. The provision extends that c-structure component to allow many more smaller trees, and thus potentially larger generation sets $Gen_{G_{\mathscr{G}}}(m)$ for some meanings. Indeed, optionality may provide a non-empty $Gen_{G_{\mathscr{G}}}(m)$ for meanings that might be inexpressible if other principles demand the presence of certain nodes or annotations.

Bresnan's definition of Economy places no constraints on the string components of the derivation triples, and hence is an instance of Different-Words Economy:

(30) Bresnan's Economy: Different-Words Economy.

Thus, her definition encompasses cases of Poser blocking, privileging (single-word) morphological over (multi-word) phrasal modes of expression of f-structures with the same content (Bresnan 2001, 93).

6.2 *Poser blocking*

Many cases of Morphological Blocking involve comparison between alternative single words in the same syntactic context, and do not fall under the purview of Economy. However, Economy is relevant for a certain subset of cases that have been treated as Blocking: Poser (1992) was among the first to explore the possibility that a slot in a morphological paradigm could be filled periphrastically, i.e., by a sequence of words, and that the availability of a means of expressing a set of features by a single word blocks the periphrastic expression of the same features. Different-Words Economy has sometimes been suggested as an explanation for these cases of morphological blocking, cases where the phrasal expression of a meaning seems to be disallowed when a single word exists that expresses the same meaning. As Nordlinger and Bresnan (2011) point out, Economy "privileges lexical over phrasal expression – morphology over syntax".[9] Thus the availability of *prettier* is claimed to block *more pretty*, whereas the non-existence of **beautifuller* is what allows for phrasal expression of the comparative of *beautiful* as *more beautiful.*

Embick and Marantz (2008) present a "generalized" formulation of Poser blocking (see also Hankamer and Mikkelsen 2002; 2005):

(31) Generalized Poser blocking (Embick and Marantz 2008, 38):

For each node in the syntactic structure, scan the lexicon for a word that expresses the same features. If such a word exists, use the word in place of the phrase.

Since comparison is over different strings – that is, single-word vs. periphrastic expression of the same meaning – string comparison in Poser blocking is an instance of Different-Words Economy. The definition in (31) can then be recast in the terms we have defined so far:

(32) Poser blocking: Different-Words Economy.

[9] This is true irrespective of whether the Economy metric counts non-preterminals or non-X^0 categories (Section 6.3), since a single X^0 category can block the expression of the same meaning by means of a larger c-structure.

There is an important difference between Embick and Marantz's interpretation of Poser blocking and Different-Words Economy: as interpreted by Embick and Marantz (2008), Poser blocking involves comparison only between single words and multi-word phrasal constituents. Although it would be formally possible to define Economy as applying only to certain subtrees in a derivation, and in particular only to pairs involving one single-word constituent and one multi-word constituent, Bresnan (2003) argues that this restriction is unsatisfactory, since it would leave a large body of data unexplained. For example, Bresnan discusses the conditional verbal paradigm in Ulster Irish (Andrews 1990), where inflected forms disallowing pronominal subjects compete with the periphrastic uninflected verb + pronominal subject, pointing out that the verb + subject in Irish do not form a constituent and so would not be involved in an Economy comparison restricted to individual subtrees in a derivation. See Bresnan (2003) for further discussion and exemplification of this point.

Treating Poser blocking as an instance of Different-Words Economy raises some important issues. In at least some cases, preference for expressing a meaning as a single word rather than periphrastically seems to be a gradient phenomenon and not a matter of grammaticality: the word *prettier* is clearly preferred (in most contexts) to the phrase *more pretty*, but the periphrastic realization may still be included in the range of expressions that the grammar allows, and in fact the periphrastic form rather than the single-word form surfaces in certain situations. Indeed, Mondorf (2009) presents an in-depth study of factors influencing synthetic vs. analytic expression of comparatives: these include number of syllables, attributive vs. predicative use, and other factors. To take just one example, Mondorf (2009, 21) gives the following counts for the comparative of the adjective *slender* in attributive, predicative, and postnominal position in a corpus comprising British newspapers and the British National Corpus:

(33)

	Synthetic	Analytic	Total	% Analytic
	(slenderer)	(more slender)		
Attributive	14	27	41	66%
Predicative	16	23	39	59%
Postnominal	3	2	5	40%
All positions	33	52	85	61%

Economy would wrongly predict that the availability of a synthetic form like *slenderer* would suppress the analytic form *more slender*; in fact, *more slender* appears in 61% of the cases overall, with *slenderer* in the remaining 39%.

In his discussion of what has come to be called Poser Blocking, Poser (1992, pp. 124–125) warns against the application of a fully general principle such as Economy to these cases, stating that

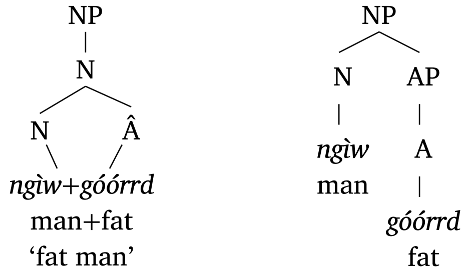
> "Under the pragmatic hypothesis, it should be possible for phrasal constructs of any size to be blocked. But in point of fact the examples of blocking of phrasal constructs known to me all involve blocking of small phrases; there appear to be no examples of blocking of large syntactic units. For example, *the red book* does not block *the book which is red.*"

Poser concludes that blocking may apply to morphological paradigms (e.g. **amn't*) but does not necessarily apply to larger syntactic units. This position was reiterated in subsequent work by Ackerman and Webelhuth (1998), Katzir (2008) and others. On this view, Poser blocking may be confined to the morphology component and should be accounted for by improved theories of periphrasis in morphology. Thus, we too believe that Economy of Expression as a general syntactic notion does not offer a proper explanation for Poser blocking.

6.3 *Nonprojecting categories and lexical sharing*

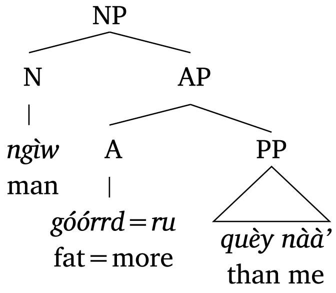
As in Toivonen's analysis of English and Swedish clitics, Economy considerations have been invoked to control whether X′ and XP levels of structure are present if they are not otherwise needed (e.g. for adjunction or coordination). Broadwell (2007) proposes to use Lexical Sharing (Wescoat 2009, 2002) and adjunction to non-projecting words to account for the distribution of Zapotec adjectives, appealing to Economy of Expression to rule out ungrammatical patterns. He points to evidence from phonology and clitic placement to show that for nouns modified by unmodified adjectives with no complements, the one-word structure in (34a) is correct and the two-word structure in (34b) is unacceptable.

(34) a. Acceptable: b. Unacceptable:



As for Swedish particles, multiword adjective phrases behave differently, and do not participate in Lexical Sharing. Adjectives with comparative complements appear as the head of a separate phrase, and do not form a single word with the noun:

(35)



'a man fatter than me'

This is similar to the Swedish patterns described by Toivonen in that separate multi-word phrases behave differently from single words, which may not form full phrases on their own; Zapotec differs from Swedish in that the adjective + noun combination forms a single word rather than a two-word sequence. The solution that Broadwell proposes is also similar: he appeals to Economy to properly discriminate between these structures, on the basis that Economy selects the smaller lexical sharing structure in (34a) to express the intended meaning, and rules out the larger structure in (34b).

Broadwell's analysis highlights an unresolved issue in the definition of Economy: which nodes are counted in determining the size of a c-structure tree? Bresnan (2001, 91) restricts attention to "syntactic phrase structure nodes", which she defines as excluding terminal and preterminal nodes: that is, to "those nonterminal nodes which do not immediately dominate a lexical element". In (34) we have adopted Bresnan's X′ Omission principle, with AP directly dominating A in example (34b). If the trees in (34) are correct, Bresnan's definition does not select the tree in (34a) over the tree in (34b). Both trees have two nonterminal nodes not dominating a lexical element, NP and N in (34a), and NP and AP in (34b), and thus should be equally economical according to Bresnan's criterion for counting nodes. If tree (34a) is to be selected on the basis of Economy, we must count non-X^0 nodes instead of non-preterminals: (34a) has only one non-X^0 node, NP, while the tree in (34b) has two non-X^0 nodes, NP and AP.

We pointed out earlier the possibility of accounting for the distribution of Swedish prepositions in terms of f-structure restrictions in the lexical entries of prepositions which optionally project, rather than an Economy-based comparison of different candidate structures. A similar constraint requiring the presence of a grammatical function may also account for the distribution of Zapotec free adjectives such as *góórrd,* but we leave details of this analysis to future research.

7 CONCLUSION

We have presented a formal framework within which explicit definitions of metagrammatical principles can be made, and we discussed three types of Economy of Expression in detail: Same-String Economy, String-Permutation Economy, and Different-Words Economy. We observed that it is important to separate the Economy metric from stylistic or pragmatic preferences that may also value succinctness or brevity. Under Economy, the only grammatical derivations for a given meaning are the smallest ones, while stylistic or pragmatic principles choose the optimal way of expressing a meaning from among grammatically well-formed derivations.

Economy as a grammatical principle is of a very different formal nature from other grammatical principles governing grammatical representations or the form of grammar rules or lexical entries: Economy requires a global choice among alternatives that are well-formed according to the other principles of the grammar. Thus, the burden of proof is on proponents of Economy to show that such a principle is necessary, and that Economy is not simply a generalization about the nature and interaction of other, independently motivated grammatical mechanisms and principles. Our view is that previous proposals have failed to provide clear motivation for an independent principle of Economy, since in all of the cases we have examined, analyses appealing to independently-motivated mechanisms provide equally good accounts of the linguistic phenomena.

Economy has been offered as a broad explanatory principle for a range of linguistic phenomena that, on close examination, do not seem to form a natural class. Our formal characterization of Economy and our survey of its empirical applications suggests that it is not a compelling explanatory principle in an LFG setting. We do not know whether other theories adopting an Economy metric have the same independently motivated mechanisms that would make Economy superfluous, but we hope our discussion has clarified some of the major issues and will help to guide further research.

ACKNOWLEDGEMENTS

Dalrymple gratefully acknowledges the support of the Leverhulme Trust (Leverhulme Research Fellowship, "Plurals: Morphology and Semantics", 2012-13, RF-2012-295). For helpful discussion, we thank Ash Asudeh, Joan Bresnan, Aaron Broadwell, Dag Haug, Sandra Kotzor, John Lowe, Louisa Sadler, Ida Toivonen, Adam Przepiórkowski, Tohru Seraku, Jürgen Wedekind, three anonymous reviewers, and the audience at LFG13, Debrecen, July 2013 and the National University of Singapore, April 2015.

REFERENCES

Farrell ACKERMAN and Gert WEBELHUTH (1998), *A Theory of Predicates*, CSLI Publications, Stanford.

Avery D. ANDREWS, III (1990), Unification and Morphological Blocking, *Natural Language and Linguistic Theory*, 8(4):507–557.

Ash ASUDEH (2006), Direct Compositionality and the Architecture of LFG, in Miriam BUTT, Mary DALRYMPLE, and Tracy Holloway KING, editors, *Intelligent Linguistic Architectures: Variations on Themes by Ronald M. Kaplan*, pp. 363–387, CSLI Publications, Stanford.

A. Machtelt BOLKESTEIN (2001), Random Scrambling? Constraints on Discontinuity in Latin Noun Phrases, in *De lingua latina, novae quaestiones: actes du Xè Colloque International de linguistique Latine*, pp. 245–258, Peeters, Louvain.

Joan BRESNAN (2001), *Lexical-Functional Syntax*, Blackwell, Oxford.

Joan BRESNAN (2003), Explaining Morphosyntactic Competition, in Mark BALTIN and Chris COLLINS, editors, *Handbook of Contemporary Syntactic Theory*, pp. 11–44, Blackwell, Oxford.

George Aaron BROADWELL (2007), Lexical Sharing and Non-Projecting Words: The Syntax of Zapotec Adjectives, in Miriam BUTT and Tracy Holloway KING, editors, *On-Line Proceedings of the LFG2007 Conference*, pp. 87–106, CSLI Publications, Stanford, `http://csli-publications.stanford.edu/LFG/12/lfg07.html`.

Chris COLLINS (2003), Economy Conditions in Syntax, in Mark BALTIN and Chris COLLINS, editors, *Handbook of Contemporary Syntactic Theory*, pp. 45–61, Blackwell, Oxford.

Mary DALRYMPLE, Ronald M. KAPLAN, and Tracy Holloway KING (2007), The Absence of Traces: Evidence From Weak Crossover, in Annie ZAENEN, Jane SIMPSON, Tracy Holloway KING, Jane GRIMSHAW, Joan MALING, and Christopher MANNING, editors, *Architectures, Rules, and Preferences: Variations on Themes by Joan W. Bresnan*, CSLI Publications, Stanford.

Mary DALRYMPLE, Ronald M. KAPLAN, John T. MAXWELL, III, and Annie ZAENEN (1995a), Formal Architecture, in Dalrymple *et al.* (1995b), pp. 1–5.

Mary DALRYMPLE, Ronald M. KAPLAN, John T. MAXWELL, III, and Annie ZAENEN, editors (1995b), *Formal Issues in Lexical-Functional Grammar*, CSLI Publications, Stanford.

Mary DALRYMPLE and Tracy Holloway KING (2013), Nested and Crossed Dependencies and the Existence of Traces, in Tracy Holloway KING and Valeria DE PAIVA, editors, *From Quirky Case to Representing Space: Papers in Honor of Annie Zaenen*, pp. 139–152, CSLI Publications, Stanford.

Mary DALRYMPLE and Louise MYCOCK (2011), The Prosody-Syntax Interface, in Miriam BUTT and Tracy Holloway KING, editors, *On-Line Proceedings of the LFG2011 Conference*, CSLI Publications, Stanford, `http://web.stanford.edu/group/cslipublications/cslipublications/LFG/16/lfg11.html`.

David EMBICK and Alec MARANTZ (2008), Architecture and Blocking, *Linguistic Inquiry*, 39(1):1–53.

Joseph EMONDS (1994), Two Principles of Economy, in Guglielmo CINQUE, Jan KOSTER, Jean-Yves POLLOCK, Luigi RIZZI, and Raffaela ZANUTTINI, editors, *Paths Toward Universal Grammar: Studies in Honor of Richard S. Kayne*, pp. 155–172, Georgetown University Press, Washington, DC.

Anette FRANK (2006), (Discourse-) Functional Analysis of Asymmetric Coordination, in Miriam BUTT, Mary DALRYMPLE, and Tracy Holloway KING, editors, *Intelligent Linguistic Architectures: Variations on Themes by Ronald M. Kaplan*, pp. 259–285, CSLI Publications, Stanford.

John FRY and Stefan KAUFMANN (1998), Information Packaging in Japanese, in Gosse BOUMA, Geert-Jan M. KRUIJFF, and Richard T. OEHRLE, editors, *Proceedings of the Joint Conference on Formal Grammar, Head-Driven Phrase Structure Grammar and Categorial Grammar (FHCG 98)*, pp. 55–65, University of the Saarlandes and DFKI, Saarbrücken.

H. Paul GRICE (1975), Logic and Conversation, in Peter COLE and Jerry MORGAN, editors, *Speech Acts*, volume 3 of *Syntax and Semantics*, pp. 43–58, Academic Press, New York, reprinted in Jackson (1991, 155–175).

Jane GRIMSHAW (2001), Economy of Structure in OT, `http://roa.rutgers.edu`, Rutgers Optimality Archive 444.

Jorge HANKAMER and Line MIKKELSEN (2005), When Movement Must Be Blocked: A Reply to Embick and Noyer, *Linguistic Inquiry*, 36(1):85–125.

Jorge HANKAMER and Line Hove MIKKELSEN (2002), A Morphological Analysis of Definite Nouns in Danish, *Journal of Germanic Linguistics*, 14(2):137–175.

Frank JACKSON, editor (1991), *Conditionals*, Oxford University Press, Oxford.

Ronald M. KAPLAN (1987), Three Seductions of Computational Psycholinguistics, in Peter WHITELOCK, Mary McGee WOOD, Harold L. SOMERS, Rod JOHNSON, and Paul BENNETT, editors, *Linguistic Theory and Computer Applications*, pp. 149–188, Academic Press, London, also published as CCL/UMIST Report No. 86.2: Alvey/ICL Workshop on Linguistic Theory and Computer Applications: Transcripts of Presentations and Discussions. Center for Computational Linguistics, University of Manchester. Reprinted in Dalrymple *et al.* (1995b, 337–367).

Ronald M. KAPLAN and Joan BRESNAN (1982), Lexical-Functional Grammar: A Formal System for Grammatical Representation, in Joan BRESNAN, editor, *The Mental Representation of Grammatical Relations*, pp. 173–281, The MIT Press, Cambridge, MA, reprinted in Dalrymple *et al.* (1995b, 29–130).

Ronald M. KAPLAN and Annie ZAENEN (1989), Long-Distance Dependencies, Constituent Structure, and Functional Uncertainty, in Mark R. BALTIN and Anthony S. KROCH, editors, *Alternative Conceptions of Phrase Structure*, pp. 17–42, University of Chicago Press, Chicago, reprinted in Dalrymple *et al.* (1995b, 137–165).

Roni KATZIR (2008), *Structural Competition in Grammar*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

Paul KAY (2002), An Informal Sketch of a Formal Architecture for Construction Grammar, *Grammars*, 5(1):1–19.

Jonas KUHN (1999), Towards a Simple Architecture for the Structure-Function Mapping, in Miriam BUTT and Tracy Holloway KING, editors, *On-Line Proceedings of the LFG99 Conference*, CSLI Publications, Stanford, http://web.stanford.edu/group/cslipublications/cslipublications/LFG/LFG4-1999/.

Britta MONDORF (2009), *More Support for More-Support: The Role of Processing Constraints on the Choice Between Synthetic and Analytic Comparative Forms*, volume 4 of *Studies in Linguistic Variation*, John Benjamins, Amsterdam.

Yukiko MORIMOTO (2001), Deriving the Directionality Parameter in OT-LFG, in Miriam BUTT and Tracy Holloway KING, editors, *On-Line Proceedings of the LFG2001 Conference*, CSLI Publications, Stanford, http://web.stanford.edu/group/cslipublications/cslipublications/LFG/6/lfg01.html.

Louise MYCOCK and John J. LOWE (2013), The Prosodic Marking of Discourse Functions, in Miriam BUTT and Tracy Holloway KING, editors, *On-Line Proceedings of the LFG2013 Conference*, CSLI Publications, Stanford, http://web.stanford.edu/group/cslipublications/cslipublications/LFG/18/lfg13.html.

Prerna NADATHUR (2013), Weak Crossover and the Direct Association Hypothesis, in Miriam BUTT and Tracy Holloway KING, editors, *On-Line Proceedings of the LFG2013 Conference*, CSLI Publications, Stanford, http://web.stanford.edu/group/cslipublications/cslipublications/LFG/18/lfg13.html.

Rachel NORDLINGER and Joan BRESNAN (2011), Lexical-Functional Grammar: Interactions Between Morphology and Syntax, in Robert D. BORSLEY and Kersti BÖRJARS, editors, *Non-Transformational Syntax: Formal and Explicit Models of Grammar*, pp. 112–140, Blackwell, Oxford.

Rachel NORDLINGER and Louisa SADLER (2007), Apposition and Coordination in Australian Languages: An LFG Analysis, *Natural Language and Linguistic Theory*, 22:597–641.

William J. POSER (1992), Blocking of Phrasal Constructions by Lexical Items, in Ivan A. SAG and Anna SZABOLCSI, editors, *Lexical Matters*, pp. 111–130, CSLI Publications, Stanford.

Paul M. POSTAL (1971), *Cross-Over Phenomena*, Holt, Rinehart & Winston, New York.

Chris POTTS (2002), Comparative Economy Conditions in Natural Language Syntax, presented at the North American Summer School in Logic, Language, and Information, Workshop on Model-Theoretic Syntax, Stanford University (June 28, 2002), http://web.stanford.edu/~cgpotts/papers/potts-nasslli-ecs.pdf.

Ivan A. SAG (2000), Another Argument Against *Wh*-Trace, in Sandy CHUNG, Jim MCCLOSKEY, and Nathan SANDERS, editors, *Jorge Hankamer Webfest*, Department of Linguistics, University of California at Santa Cruz.

Ivan A. SAG and Janet D. FODOR (1994), Extraction Without Traces, in Raul ARANOVICH, William BYRNE, Susanne PREUSS, and Martha SENTURIA, editors, *WCCFL 13: Proceedings of the 13th West Coast Conference on Formal Linguistics*, pp. 365–384, CSLI Publications, Stanford.

Liselotte SNIJDERS (2012), Issues Concerning Constraints on Discontinuous NPs in Latin, in Miriam BUTT and Tracy Holloway KING, editors, *On-Line Proceedings of the LFG2012 Conference*, CSLI Publications, Stanford, `http://web.stanford.edu/group/cslipublications/cslipublications/LFG/17/lfg12.html`.

Ida TOIVONEN (2002), Verbal Particles and Results in Swedish and English, in *WCCFL 21: Proceedings of the 21st West Coast Conference on Formal Linguistics*, Cascadilla Press, Medford, MA.

Ida TOIVONEN (2003), *Non-Projecting Words: A Case Study of Swedish Verbal Particles*, Kluwer, Dordrecht.

Thomas WASOW (1979), *Anaphora in Generative Grammar*, E. Story-Scientia, Ghent.

Jürgen WEDEKIND and Ronald M. KAPLAN (2012), LFG Generation by Grammar Specialization, *Computational Linguistics*, 38(4):867–915.

Michael T. WESCOAT (2002), *On Lexical Sharing*, Ph.D. thesis, Stanford University.

Michael T. WESCOAT (2009), Udi Person Markers and Lexical Integrity, in Miriam BUTT and Tracy Holloway KING, editors, *On-Line Proceedings of the LFG2009 Conference*, pp. 604–622, CSLI Publications, Stanford, `http://web.stanford.edu/group/cslipublications/cslipublications/LFG/14/lfg09.html`.

Complex predicates: an LFG+glue analysis

John J. Lowe
University of Oxford

ABSTRACT

In this paper I discuss weaknesses in the traditional LFG account of complex predicates and in the XLE implementation of the same. I argue that the concept of predicate composition in general, and the mechanisms required to achieve it, are problematic, but that the most problematic element is the concept of argument fusion. I show that a semantically-integrated account of complex predicate formation is possible within LFG + glue, an account which provides a simple and effective formalization of argument fusion, and which does not suffer from the weaknesses of traditional approaches.[1]

Keywords: complex predicates, Lexical Functional Grammar, argument structure, glue semantics

1 INTRODUCTION

Complex predicates present a challenge to any lexicalist theory of syntax since, in at least some languages, there is clear evidence that a single clausal predicate can result from a *syntactic* process involving two or more distinct lexical elements (usually a lexical verbal or nominal element, and one or more 'light' verbal elements). The resulting

[1] I am very grateful to the attendees of the Oxford Glue Group, 8 May 2015, and the audience at the 18th SE-LFG meeting, SOAS, 31 October 2015, where versions of this work were presented, for their attention and valuable comments, in particular to Mary Dalrymple, Avery Andrews, Jamie Findlay, Louisa Sadler, Kersti Börjars, Andrew Spencer, and Peter Austin. I am also grateful to Anna Kibort, whose paper on causatives at the 16th SE-LFG meeting, SOAS, 21 February 2015, first got me thinking about complex predicates and glue. Naturally all errors are my own. This work was undertaken while I was in receipt of an Early Career Research Fellowship funded by the Leverhulme Trust.

predicate functions as if it were a single lexical element, but its formation within the syntax belies this. In this paper I discuss a number of approaches to complex predicate formation within the strict lexicalist theory of Lexical Functional Grammar (LFG; Kaplan and Bresnan 1982; Bresnan 2001; Falk 2001), and show that all suffer from theoretical, and in some cases even empirical, weaknesses. I then present an analysis within LFG augmented with glue semantics (LFG + glue; e.g. Dalrymple 2001; Asudeh 2012), which overcomes the weaknesses in previous approaches and even has the potential to account for data which is problematic for previous accounts.

Early work on complex predication within LFG proposed either a multiclausal syntactic analysis similar to raising (e.g. Ishikawa 1985), or an essentially lexical analysis, whereby complex predicates are formed from their constituent parts inside the lexicon (e.g. Kaplan and Wedekind 1993; Ackerman and Webelhuth 1996, 1998). However, authors such as Mohanan (1994), Butt (1995) and Alsina (1996) demonstrated beyond reasonable doubt that some languages attest complex predicates which are syntactically monoclausal, yet must be analysed as formed in the syntax. From an LFG perspective, the challenge in modelling such a phenomenon lies in the process of predicate formation, in particular in the merger of distinct semantic forms, since semantic forms are in principle not manipulable in syntax, and in the fusion and linking of the arguments of merged predicates. Since the early work of Butt (1995) and Alsina (1996), there has been a wealth of research on complex predicate formation as a syntactic phenomenon within LFG, in particular by Miriam Butt and her colleagues.[2] Two main formal approaches have developed: one that now might reasonably be called the 'traditional' LFG approach, which seeks to integrate the analysis of complex predicates with work on argument structure and 'linking theory', and a somewhat different approach which is utilized in the computational implementation of LFG, XLE (Crouch *et al.* 2011). Relatively little work has been done, however, on how semantics interacts with the syntax and argument structure of complex predicate formation; the exceptions are Kaplan and Wedekind (1993), Dal-

[2] See e.g. Butt (1997), Butt and Geuder (2001), Butt *et al.* (2003), Butt and Ramchand (2005), Butt and King (2006), Butt *et al.* (2010), Ahmed and Butt (2011), Raza (2011), Ahmed *et al.* (2012), Butt *et al.* (2012), Sulger (2012), and Butt (2014).

rymple *et al.* (1993a), Andrews and Manning (1999), Andrews (2007), and Homola and Coler (2013).[3] In particular, there exists no account of complex predicates within standard architectural assumptions and in the current standard 'new' glue format. Recent work in LFG + glue, e.g. by Asudeh and Giorgolo (2012), has shown that glue semantics is able to do a lot of the work traditionally attributed to argument structure; one aim of this paper is to show that this holds also for complex predication.

In the next section I show that neither of the main approaches to complex predicate formation in LFG provides an entirely satisfactory analysis of predicate composition or argument merger. In §3 I argue that a semantically integrated account is more satisfactory; in §4 I show that my proposal can not only deal with some of the most complex phenomena that previous accounts can, but that it even has the potential to deal with phenomena that are problematic for previous accounts. In §5 my proposal is compared with previous proposals for a semantic account of complex predicates in LFG. In §6 I draw my conclusions.

2 THE STANDARD ACCOUNTS

As mentioned in the previous section, there are two approaches that might be considered the standard approaches to complex predicates in LFG. This is not to say that there are two *competing* approaches, or that it is a case of some authors advocating one approach over the other. Rather, the two approaches are used in different contexts, even by the same authors. For example, Butt (2014) provides one of the most elegant and fully formulated accounts of what I will refer to as the 'linking' approach, which builds on much of her previous work, but at the same time Butt has been at the forefront of developing

[3] Current work in XLE does not attempt to integrate glue semantics, or any theory of the syntax-semantics interface, into the implementation. Functional means of dealing with semantic representations are available, by means of the f-structure LEX-SEM feature or by means of f-structure rewriting (Crouch and King 2006), but these permit no active role for semantics in the grammar. The absence of a semantically integrated account of complex predicates within XLE does not therefore have anything specifically to do with complex predicates but is merely a feature of the XLE implementation at the present time.

the XLE treatment of complex predicates within the context of the Urdu PARGRAM grammar (Butt *et al.* 1999, 2002; Butt and King 2007; Sulger *et al.* 2013).

The very fact that the computational implementation of LFG does not include a full formalization of the linking approach raises something of a question mark over both approaches, in particular over the lack of formalization of the linking approach, and over the analytical accuracy of the XLE approach.[4] In this section both approaches are described, focusing initially on those aspects that both approaches share, and then drawing out the ways in which they differ. The description of the linking theory approach is based on the recent account of Butt (2014).

The phenomenon in question is exemplified in (1) and (2):[5] (1) shows a simple transitive sentence in Urdu with the verb *likh* 'write', while (2) shows a sentence involving a complex predicate formed of the verb *likh* 'write' and the 'permissive' light verb *de*.[6]

(1) *saddaf-ne* *ciṭṭhii* *likh-ii*
Saddaf-ERG note.NOM.F.SG write-PERF.F.SG
'Saddaf wrote a note.' (Urdu)

(2) *anjum-ne* *saddaf-ko* *ciṭṭhii* *likh-ne*
Anjum-ERG Saddaf-DAT note.NOM.F.SG write-INF.OBL
d-ii
let-PERF.F.SG
'Anjum let Saddaf write a note.' (Urdu)

As Butt and other authors have demonstrated, Urdu complex predicates such as that in (2) are monoclausal at f-structure but consist of two predicating elements, each with their own argument structures. Light verbs can combine productively and recursively with most verbal, and many nominal, forms, such that their combination must be treated syntactically, not lexically.

[4] The reasons for the differences between the two approaches are discussed by Butt *et al.* (2010, 249–250); they boil down to the desire for computational efficiency within XLE.

[5] The examples are from Butt (2014).

[6] The following abbreviations are used in the glosses: CAUS 'causative', DAT 'dative', ERG 'ergative', F 'feminine', INF 'infinitive', INSTR 'instrumental', M 'masculine', NOM 'nominative', OBL 'oblique', PERF 'perfect', SG 'singular'.

Under the linking approach to complex predicates, the lexical entry for the verb *likh* 'write' is assumed to contain the semantic form specification in (3), while the lexical entry for the light verb *de* 'let' is assumed to contain the semantic form specification in (4).

(3) $$(\uparrow \textsc{pred}) = \text{`write } \langle\ \underset{[-\textsc{o}]}{\textsc{agent}},\ \underset{[-\textsc{r}]}{\textsc{theme}}\ \rangle\text{'}$$

(4) $$(\uparrow \textsc{pred}) = \text{`let } \langle\ \underset{[-\textsc{o}]}{\textsc{agent}},\ \underset{[+\textsc{r}]}{\textsc{goal}},\ \%\textsc{pred}\ \rangle\text{'}$$

In these semantic forms, the verb forms concerned subcategorize for arguments which are defined by reference to the semantic role of the argument and by reference to one of the features ±O or ±R, which constrain the mapping between semantic roles and grammatical functions according to the principles of Mapping Theory (Bresnan and Kanerva 1989). The specifics of the argument structure model assumed, and the details of Mapping Theory, are not important for the present purposes; the representations of Butt (2014) are adopted here, but e.g. all the semantic forms and argument structure representations presented in this paper could easily be rewritten in the model of Kibort (2001, 2004, 2006, 2007, 2008), and no significant differences would result.

What is important is that these semantic forms must fuse in the formation of the f-structure, with the semantic form of the lexical verb supplying the value of the %PRED variable in the argument structure of the light verb. This process of fusion is discussed in more detail in the rest of this section; at this point it suffices to say that the selected semantic roles are associated with grammatical functions, and that a single predicate, with a single subcategorization frame, results. This can be seen in the PRED value in (5), which shows the resulting f-structure for the clause in (2).

(5) $$\left[\begin{array}{ll} \textsc{pred} & \text{`let-write}\langle\textsc{subj},\ \textsc{obj}_{\text{goal}},\ \textsc{obj}\rangle\text{'} \\ \textsc{subj} & \left[\textsc{pred}\ \ \text{`Anjum'}\right] \\ \textsc{obj}_{\text{goal}} & \left[\textsc{pred}\ \ \text{`Saddaf'}\right] \\ \textsc{obj} & \left[\textsc{pred}\ \ \text{`note'}\right] \end{array}\right]$$

The first theoretical weakness of the linking theory approach, a feature shared with the XLE approach, is in the mechanism of predicate fusion. A fundamental assumption of early lexicalist syntax was the principle of Direct Syntactic Encoding, i.e. the principle that lexical properties such as argument structure should not be manipulable in the syntax. This plays out in LFG in the fact that, at least originally, semantic forms are not manipulable in the syntax. As clearly demonstrated by Mohanan (1994), Butt (1995) and Alsina (1996), however, complex predicates require a syntactic explanation, and in this respect, at least, the principle of Direct Syntactic Encoding cannot be maintained. Under the linking and XLE approaches to complex predication an exception to the non-manipulability of semantic forms must be made, since there is no other way for predicates to compose, and the variable %PRED utilized in the semantic forms for light verbs (as in (4)) was adopted as a means of manipulating semantic forms outside the lexicon. The variable %PRED is therefore an augmentation of the original LFG system which, though apparently necessary, significantly increases its power, and is required purely to account for complex predicates. If %PRED, and manipulable semantic forms in general, could be eliminated, this would be theoretically advantageous in restricting the power of the LFG formalism and reducing the number of construction-specific devices required.

A further problem with predicate fusion is the mechanism required to actually get the information supplied by the embedded semantic form inside the semantic form of the light verb, i.e. precisely how a semantic form such as that in (6) gets instantiated as (7).[7]

(6) ‘let ⟨ AGENT, GOAL, %PRED ⟩’

(7) ‘let ⟨ AGENT, GOAL, ‘write ⟨ AGENT, THEME ⟩’ ⟩’

Most recent discussions of complex predication that are based within the linking approach brush over the explicit formalization of this process. In early work, Butt (1995) and Alsina (1996, 1997) do provide formalized accounts of the process. Butt’s (1995) account ne-

[7] An instantiation as in (7) is usually represented in an f-structure in resolved form, that is with a single ‘fused’ predicate with a single subcategorization frame, and with subcategorization for semantic roles replaced by subcategorization for grammatical functions, as shown in (5).

cessitates assuming a distinction between two types of semantic form, one type (found with light verbs) which is incomplete on its own and requires that it be unified with a standard, complete, semantic form. In addition, the usual $\uparrow = \downarrow$ f-description must be reinterpreted such that it licenses the composition of semantic forms where necessary. Alsina's (1997) proposal is similar, except that the alternative interpretation of $\uparrow = \downarrow$ is associated with a new function $\uparrow =_H \downarrow$, and the precise formulation of the composition is stated in somewhat different terms. Both accounts involve augmentations of the standard LFG model, thereby increasing its power and, as argued by Andrews and Manning (1999), the proposals are either under-formalized in certain respects, or else there are difficulties with the formalizations involved. In any case, neither proposal appears to have been widely adopted, at least explicitly, in recent work within the linking approach.

Besides these early proposals, the only remaining available formalization is that proposed by Butt *et al.* (2003) for XLE, and in the following I assume that this formalization holds also for the linking approach.[8] For the XLE approach, it is necessary to assume that semantic forms can be decomposed into their constituent parts. In particular, the feature ARG$_x$ can be used to refer to argument positions inside the PRED feature. That is, for example, the constituent parts of the semantic form in (6) can be referred to by the schema:

(8) 'FN $\langle$ ARG$_1$, ARG$_2$, ARG$_3$ $\rangle$'

such that for any f-structure for which (6) provides the PRED, the %PRED variable can be referred to by the path PRED ARG$_3$. Then, via a phrase structure rule such as that in (9), the PRED of a lexical verb can be identified with the %PRED slot in the PRED value of a light verb.[9]

[8] Butt *et al.* (2010) discuss the following details, in particular the use of the restriction operator, as specifically part of the XLE approach to complex predicates and not as part of the linking approach. However, as stated, no standard formalization exists for the linking approach (Butt *et al.* make no mention of what they assume) such that, to the extent that one wants to be able to formalize predicate fusion in the linking approach, one is essentially constrained to make use of the XLE mechanisms.

[9] This rule has been simplified for the purposes of exposition; a more detailed version is given in (11).

(9)
$$\begin{array}{llcc} \text{V} & \rightarrow & \text{V}_{\text{lex}} & \text{V}_{\text{light}} \\ & & \downarrow\backslash\textsc{pred} = \uparrow\backslash\textsc{pred} & \uparrow=\downarrow \\ & & (\uparrow \textsc{pred arg}_3) = (\downarrow \textsc{pred}) & \end{array}$$

This works, but it suffers from the same problem that we have seen already with regard to the %PRED variable: the $\textsc{arg}_{\text{x}}$ feature is required specifically to account for predicate composition, and its purpose is to enable the manipulation of an otherwise non-manipulable element of f-structure, the semantic form.[10] Furthermore, this analysis must make use of the restriction operator \, as seen in (9). The restriction operator was introduced by Kaplan and Wedekind (1993), who provide the following definition:

(10) If f is an f-structure and a is an attribute:
$$f\backslash a = f|_{\text{Dom}(f)-\{a\}} = \{<s,v>\in f|s \neq a\}$$

Informally, the f-structure $f\backslash a$ is identical to the f-structure that results from removing the attribute a from the f-structure f. This operation is a fundamental part of both the linking theory and XLE analyses of complex predication, since they seek to represent the fact that both lexical and light verb elements are co-heads of the clausal f-structure, even though some attributes of the clausal f-structure have different values from those required by the lexical verb. One of these attributes is PRED, as seen in (9): the PRED of the lexical verb's f-structure serves as an argument inside the PRED of the light verb (and thereby the clause), so the two are necessarily not the same. In the simplified phrase-structure rule given in (9), only one restriction is stated, but full treatments require considerable use of restriction. For example, Butt *et al.* (2003, 99) provide the following rule for complex predication in Urdu (explicitly for the XLE approach):

(11)
$$\begin{array}{llcc} \text{V} & \rightarrow & \text{V} & \text{V}_{\text{light}} \\ & & \downarrow\backslash\textsc{pred}\backslash\textsc{subj}\backslash\textsc{vtype}\backslash\textsc{lex-sem} = & \\ & & \uparrow\backslash\textsc{pred}\backslash\textsc{subj}\backslash\textsc{obj}_{\text{goal}}\backslash\textsc{vtype}\backslash\textsc{lex-sem} & \uparrow=\downarrow \\ & & (\uparrow \textsc{pred arg}_3) = (\downarrow \textsc{pred}) & \\ & & (\uparrow \textsc{obj}_{\text{goal}}) = (\downarrow \textsc{subj}) & \end{array}$$

[10] The FN feature has found more widespread use, but both are rendered unnecessary for any phenomenon under the proposals made in §3.

Restriction is a well-defined set-theoretic operation, and is not in principle to be avoided. Bresnan apud Butt *et al.* (2010, 253) questions the use of the restriction operator on theoretical grounds, since it potentially endangers the Principle of Direct Syntactic Encoding by permitting grammatical functions to be changed in the syntax; this is really the same problem we have seen already with the other aspects of the formalization of predicate composition. A more specific problem is that it may cause inside-out functional uncertainty to fail (Andrews 2001, and p.c.).[11] In any case, an analysis that can dispense with restriction is perhaps to be preferred over one that cannot do so purely on grounds of simplicity.

In fact, the use of restriction has some not entirely desirable consequences. The intuition behind the use of the restriction operator here is, as mentioned, that both the lexical verb and the light verb are coheads of the clausal f-structure. This is a key part of the important observation that such complex predicates are monoclausal at f-structure. Nevertheless, while equations of the type $\downarrow \backslash\textsc{pred} = \uparrow \backslash\textsc{pred}$ do in some sense permit the lexical verb to function as a co-head, they also specify the existence of a separate f-structure, of which the lexical verb alone is the head. That is, e.g., for the sentence in (2), alongside the f-structure in (5) there must also exist that in (12), which represents the f-structure for the lexical verb alone.

(12)
$$\begin{bmatrix} \textsc{pred} & \text{‘write}\langle\textsc{subj}, \textsc{obj}\rangle\text{’} \\ \textsc{subj} & \begin{bmatrix}\textsc{pred} & \text{‘Saddaf’}\end{bmatrix} \\ \textsc{obj} & \begin{bmatrix}\textsc{pred} & \text{‘note’}\end{bmatrix}\end{bmatrix}$$

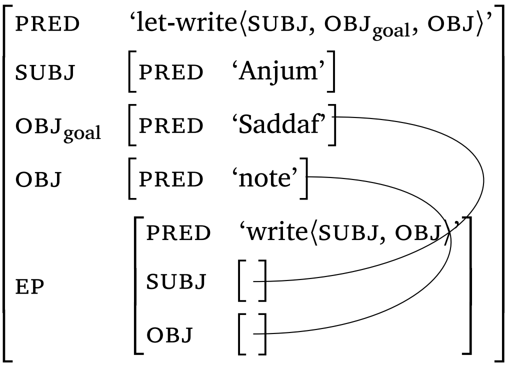
Butt *et al.* (2003, 101) refer to the full clausal f-structure of a complex predicate, such as that in (5), as representing “the final analysis”, implying that the separate f-structure for the lexical verb is somehow preliminary and not independently part of the final analysis. However, by the phrase-structure rules and f-descriptions that specify both clausal and lexical verb f-structures, there is no sense in which one f-structure is in any sense subordinate to, or subsumed or rendered

[11] Recent work by Homola and Coler (2013) sets out explicitly to eliminate the need for restriction in the analysis of complex predicates; I will discuss this in more detail below.

superfluous by, the other. Both exist, side by side, sharing all features not restricted out, but potentially differing in respect of the restricted features. This means that, although it is a fundamental assumption of the linking and XLE approaches to complex predicates that the lexical verb – light verb complex is monoclausal at f-structure, the only widely utilized and fully formalized analysis of this in LFG requires that there are in fact two f-structures (contrary to the original analyses of Butt 1995 and Alsina 1996, 1997). It is worth remarking that the only real value of the restriction operator here is to permit these two f-structures to exist side by side, rather than one embedded inside the other. That is, if one were prepared to permit the f-structure for the lexical verb to be embedded inside the f-structure for the clause, it would in principle be possible to do away with the restriction operator. For example, a phrase-structure rule such as that in (13) would produce an f-structure such as (14) for the sentence in (2).[12]

(13)
$$\mathrm{V} \quad \rightarrow \quad \begin{array}{c} \mathrm{V} \\ (\uparrow \textsc{ep}) = \downarrow \\ (\uparrow \textsc{pred arg}_3) = (\downarrow \textsc{pred}) \\ (\uparrow \textsc{obj}_{\mathrm{goal}}) = (\downarrow \textsc{subj}) \\ (\uparrow \textsc{obj}) = (\downarrow \textsc{obj}) \end{array} \quad \begin{array}{c} \mathrm{V}_{\mathrm{light}} \\ \uparrow = \downarrow \\ \\ \\ \\ \end{array}$$

(14)



The similarity with a raising analysis of complex predication is obvious. But, as stated, the fundamental assumption of these approaches is that a raising-like analysis involving a multiclausal f-structure is not appropriate, since there is very good evidence for monoclausality.

[12] As in (5), with the PRED value shown in resolved, i.e. 'fused', form.

However, as long as the lexical verb's f-structure is not directly subcategorized for by the light verb (hence the use of the ad hoc EP in (14), standing for 'embedded predicate', rather than e.g. COMP), and as long as all the arguments of the complex predicate appear in the clausal f-structure by virtue of the identification of the lexical verb's PRED with an argument of the light verb's PRED, it could be argued that the evidence for monoclausality does not in principle exclude the embedding of an f-structure for the lexical verb inside the clausal f-structure. That is, if the outer f-structure in (5) is f, the outer f-structure in (14) is g, and the attribute EP is e, then $f = g \backslash e$, and any evidence for monoclausality can be explained by reference to $g \backslash e$ just as easily as it can by reference specifically to f. In other words, if the lexical verb must head its own f-structure, it does not really matter whether this f-structure appears inside the clausal f-structure, as in (14), or alongside it, as the linking/XLE analyses assume. Details aside, embedding is essentially the approach taken by Andrews and Manning (1999), whose proposals involve the embedded predicate appearing as the value of an f-structure feature ARG.

Given the evidence for monoclausality, it would be preferable if the analysis could eliminate the need for a separate f-structure headed by the lexical verb altogether. As discussed, the f-structure for the lexical verb in a complex predicate is not treated as part of the "final analysis". The assumption of such an f-structure is, in terms of the syntax, at least, little more than a technical necessity for the linking/XLE approaches to be able to account for predicate composition. On the other hand, there may be *semantic* difficulties with assuming only a single f-structure. This is discussed in more detail in §4, but at this point one may draw the conclusion that if multiple f-structures are necessary, there seems to be little gained by using restriction when all it achieves is disconnecting those f-structures from one another.

Thus far, I have avoided detailed discussion of the arguments of complex predicates. Beside the process of predicate, and f-structure, composition, this is the second major question mark over the linking/XLE analyses of complex predication in LFG. It is also apparently the most problematic, since while there do exist formal accounts of predicate composition within LFG (however problematic), there exists no comparable formalization of argument fusion. Although it is an aim of this paper to provide a general treatment of complex predicates, the primary aim is to show that the hitherto unformalized process of argument fusion receives a formally elegant account when treated within LFG + glue.

The issue is how in (7), for example, the AGENT argument of the lexical verb *likh* ‘write’ is fused with the GOAL argument of the light verb, such that only the single resulting fused argument undergoes mapping to a grammatical function (i.e. with the result that there are only three arguments of the complex predicate ‘let-write’, rather than four; cf. (2) and (5)). At this point, the linking and XLE approaches go their separate ways. As for the linking approach, there has been considerable work on the argument structure relations involved, and how the arguments resulting from the fusion of two predicates map correctly to their respective grammatical functions. Generalizations have also been stated on which arguments may fuse: e.g. Butt (1995, 1998) proposes that the lowest matrix argument must be identified with the highest embedded argument. In terms of the actual process of argument fusion, however, I am aware of no explicit account within the linking approach, even in the most recent work by Butt (e.g. 2014). As for XLE, argument fusion is simply avoided.

In (1), *saddaf-ne* is the agent and the subject (or [−O] argument in linking theory terms). However, in (2), the equivalent argument is still the agent of the event of writing, but it now surfaces as OBJ_{goal} in the f-structure. That is, the argument structures for the predicates of the two examples are respectively:

(15)	‘write’ ⟨	AGENT	THEME ⟩
		[−O]	[−R]
		|	|
		SUBJ	OBJ

(16)	‘let’ ⟨	AGENT	GOAL	‘write’ ⟨	AGENT	THEME ⟩⟩
		[−O]	[+O]		([−O])	[−R]
		|	|			|
		SUBJ	OBJ_θ			OBJ

The problem is how the AGENT of the lexical verb is fused with the GOAL argument of the light verb, resulting in an argument that maps to OBJ_θ. In the case in question the fused argument adopts the properties of the light verb’s argument: it adopts the [+O] of the light verb’s

GOAL argument, and not the [−O] of the lexical verb's AGENT, meaning that it can map to an object function (here OBJ_θ). As stated, in linking approaches to complex predication, there is no explicit account of how this happens, even though it is a fundamental element of the approach.

That the argument fusion assumed in the linking approach is a badly underformalized notion is evident from the fact that, as mentioned, the XLE approach is rather different. In XLE there is no such thing as argument fusion. While linking accounts of complex predication consistently assume that a light verb such as Urdu *de* 'let' (and similar light verbs, such as causatives) is a three-place predicate, subcategorizing for two thematic arguments (for *de* an AGENT and a GOAL) and one predicate argument, in XLE such light verbs are two-place, subcategorizing for only one thematic argument and one predicate argument. For example, in XLE the lexical entry for Urdu permissive *de* will include the following (Butt *et al.* 2003, 99):

(17) (↑ PRED) = 'let⟨(↑ SUBJ), %PRED2⟩'

Since the light verb here introduces only one thematic argument, for a complex predicate such as 'let-write' there is no need for argument fusion, since only three thematic arguments are introduced by the separate verbs: one by the light verb, two by the lexical verb. All that is needed is for the grammatical function of the lexical verb's SUBJ to be changed as appropriate when it appears in the clausal f-structure; this is achieved by f-descriptions such as (↑ OBJ_{goal}) = (↓ SUBJ), as seen in (11).[13] That this is very different from the linking approach to complex predicates is noted by Butt and King (2006), who point out that the XLE approach is closer to some Minimalist analyses of complex predication.

A further feature of both the linking and XLE approaches to complex predication is that neither involves an explicit account of the se-

[13] It is a further weakness of the XLE approach that this f-description has to appear as an annotation in the phrase-structure rule under the lexical verb's V, rather than under the light verb's V. In principle, one would expect the specification to be associated with the light verb; indeed, the grammatical function of the argument depends on the light verb, since while e.g. Urdu *de* 'let' requires the lexical verb's SUBJ to appear as OBJ_θ, another light verb might require it to appear as an OBL_θ.

mantic aspect.[14] In the following section, I develop an alternative approach to complex predication, which makes use of glue semantics not only to provide a proper semantic analysis of complex predication, but also to overcome the weaknesses of the linking/XLE approaches.

3 PROPOSAL

It has long been recognized that the resource sensitivity of glue semantics has the potential to capture a number of constraints that must otherwise be dealt with at other levels of structure.[15] In particular, the resource sensitivity of glue means that the principles of COMPLETENESS and COHERENCE, traditionally treated as well-formedness constraints on f-structure, are captured at the level of semantics, rendering them superfluous as f-structure constraints. This means the subcategorization frame traditionally assumed as part of an f-structure PRED feature is unnecessary: subcategorization can be dealt with almost entirely within the semantics (Kuhn 2001; Asudeh 2004, 2012; Asudeh and Giorgolo 2012).[16]

Therefore, the subcategorization requirements of a complex predicate, and the process of argument fusion, however understood, can be dealt with in the semantic representation. This permits an immediate simplification of the syntactic representation: it is no longer necessary to assume predicate composition in the f-structure, since the purpose of predicate composition is essentially to enable the combination of the subcategorization requirements of both the lexical verb and the light verb in the same PRED feature. At a stroke, manipulable PRED features, the %PRED variable, and the restriction operator are all rendered unnecessary. So in place of the phrase structure rule in (9), it is sufficient for the present purposes to assume the phrase structure rule in (18) for complex predicates in Urdu.

(18) $$V_{(\text{lex})} \rightarrow \underset{\uparrow=\downarrow}{V_{\text{lex}}} \quad \underset{\uparrow=\downarrow}{V_{\text{light}}}$$

[14] Cf. fn. 3.

[15] The earliest recognition of this may be by Kaplan apud Dalrymple *et al.* (1993a, 14); see also Kuhn (2001) and Andrews (2008).

[16] Non-semantic arguments cannot be dealt with in the semantics, but they can still be handled without recourse to subcategorization in semantic forms.

That is, the lexical verb and light verb are genuine co-heads, reflecting the original intuition regarding the construction. Under this analysis, if we wish to avoid the difficulties of predicate composition, only one verb can supply a PRED value. Since complex predicates can be recursively embedded under light verbs to form new predicates, the PRED must be supplied by the lexical verb. Light verbs may then contribute only features.[17] For example, rather than the f-structure in (5), I assume for the present an f-structure as in (19), based on lexical specifications as in (20) and (21).[18]

(19)
$$\begin{bmatrix} \textsc{pred} & \text{`write'} \\ \textsc{permissive} & + \\ \textsc{subj} & \begin{bmatrix} \textsc{pred} & \text{`Anjum'} \end{bmatrix} \\ \textsc{obj}_{\text{goal}} & \begin{bmatrix} \textsc{pred} & \text{`Saddaf'} \end{bmatrix} \\ \textsc{obj} & \begin{bmatrix} \textsc{pred} & \text{`note'} \end{bmatrix} \end{bmatrix}$$

(20) *likh*: $(\uparrow \textsc{pred}) = \text{`write'}$

(21) *de*: $(\uparrow \textsc{permissive}) = +$

The point is that once subcategorization is removed from semantic forms, and given that the existence of a separate semantic representation eliminates any requirement for semantic forms to reflect semantic content, a light verb need contribute no more than (and perhaps not even as much as) an f-structure feature specifying permission, or

[17] In fact, they need not even contribute features, if there are no syntactic operations that would require reference to such features. In the analysis proposed here it is assumed that Urdu light verbs do contribute features, but this is assumed largely to make the f-structure representations clearer, and I make no firm claims as to whether such features are strictly necessary.

[18] The PRED value and subcategorized grammatical functions are not the only features that necessarily show different values for the lexical and light verb. In (11), for example, one of the features restricted out is VTYPE, since the light verb is finite and the lexical verb an infinitive. The solution for any such feature will depend on the function that it has in the wider grammar, but none should be impossible to deal with. In the case of VTYPE, for example, it would be possible to deal with this at 'morphological structure' (Butt *et al.* 1996, 1999), i.e. in just the same way as monoclausal auxiliary sequences in English.

causation, etc.: it no longer needs to contribute anything to the clausal PRED itself.[19]

This observation is not, in fact, new. Dalrymple *et al.* (1993a) point out that:

> If the only remaining function of the PRED is to ensure predicate uniqueness, it would do as well to assume that the PRED value for a sentence with a complex predicate is contributed by the main verb…, and that the function of [a light verb such as] LET is to modify the argument structure but not to contribute to or change the PRED value of the construction.

Dalrymple *et al.* (1993a) still assume complex PRED features of the form 'PERMIT⟨WRITE⟩', but they make no claims as to how they would be formed, and they assume such features perhaps only for the sake of greater consistency with existing accounts. Nevertheless, Dalrymple *et al.*'s important insight has been essentially ignored in work in both the linking and XLE approaches (presumably because these approaches tend to lack an explicit semantic angle), and it is well worth re-emphasizing.

At least superficially, the problematic concept of 'argument fusion' is more difficult to address, and it is here that the value of a glue-based approach becomes apparent. The problem essentially boils down to the question of how arguments are recategorized when they appear inside a complex predicate. Assuming that subcategorization is not dealt with in the f-structure, but only in the semantics, let us consider how a very simple glue semantic account might fare. A standard glue treatment of verbal meaning might assume the following meaning constructor for *likh* 'write' (assuming a very simple event semantics, making use of an event variable

[19] That is, following the Dalrymple *et al.* quote provided, I assume that the only important property of PRED features is their unique instantiation, which serves to distinguish any two f-structures that have PRED features; the value itself is unimportant. Thus it does not matter that the PRED value in (19) does not reflect the meaning of the complex verb (since the value is 'write' but the meaning of the full predicate is 'let write'). This was relevant only in pre-glue LFG, but is superfluous in LFG + glue, since semantic content is represented separately from the f-structure. What function PRED values do serve in LFG + glue is a matter for debate; see Andrews (2008) for discussion.

but ignoring temporal variables usually assumed in more elaborate treatments of event semantics in glue, e.g. Fry 2005, Haug 2008, Lowe 2015):

(22) $\lambda y.\lambda x.\lambda e.write(e) \wedge agent(e,x) \wedge theme(e,y) : (\uparrow \textsc{obj})_\sigma \multimap (\uparrow \textsc{subj})_\sigma \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma$

Meaning constructors such as this render subcategorization in the f-structure, and thereby also completeness and coherence as f-structure well-formedness constraints, superfluous, since the glue expression ensures that only a SUBJ and an OBJ, and no other feature, can and must appear as governable grammatical functions in the f-structure headed by the verb, else an incoherent semantics would result. But what this meaning constructor also does is effectively tie the agent of the event of writing to the grammatical function SUBJ, and the theme of the event of writing to the grammatical function OBJ. This is fine for the simplex verb, but when embedded under a complex predicate the SUBJ should not be the agent of the event of writing: it will either have no thematic relation to the event of writing, or a relation of 'permitter', depending on how we choose to model the semantics of the light verb.

Asudeh and Giorgolo (2012) and Asudeh *et al.* (2014), in their glue-based approach to argument structure and valency alternations, propose meaning constructors of slightly different form from the sort in (22), but the basic problem is the same. In their approach, f-structural grammatical functions such as SUBJ and OBJ are linked with s-structure features labelled $\textsc{arg}_1$, $\textsc{arg}_2$, etc., via f-descriptions in the lexical entries of verbs.[20] So, the equation in the third line of the lexical entry in (23) identifies the semantic structure projected from the verb's SUBJ with an s-structure labelled $\textsc{arg}_1$ in the s-structure projected from the verb's f-structure. Then, the glue expressions in the meaning constructor for the verb make reference to the s-structure features $\textsc{arg}_1$, etc., and do not make direct reference to grammatical functions like SUBJ.

[20] S-structure features $\textsc{arg}_1$ etc. are not related to the f-structure $\textsc{arg}_x$ features discussed in §2.

(23) 'write' V
$(\uparrow \textsc{pred}) = \text{`write'}$
$(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg}_1)$
$(\uparrow \textsc{obj})_\sigma = (\uparrow_\sigma \textsc{arg}_2)$

$\lambda y.\lambda x.\lambda e.write(e) \wedge agent(e,x) \wedge theme(e,y):$
$(\uparrow_\sigma \textsc{arg}_2) \multimap (\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma$

Nevertheless, from the current perspective, it remains the case that e.g. $\textsc{arg}_1$ in the meaning constructor in (23) is tied to the agent of the act of writing (y on the meaning side), and by the equation $(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg}_1)$ this is tied to the grammatical function SUBJ.

The problem remains also in Findlay's (2014) fusion of Asudeh and Giorgolo's (2012) proposals with Kibort's (2001; 2004; 2006; 2007; 2008) model of argument structure (briefly detailed in Asudeh *et al.* 2014, 75–77), even though Findlay's model provides for greater flexibility in the association between grammatical functions and s-structure $\textsc{arg}_x$ features. In Findlay's model, the equation $(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg}_1)$ in (23) would be replaced by the following equation (adopting the notation of Asudeh *et al.* 2014, 76 and omitting the use of templates):

(24) $\{(\uparrow \{\textsc{subj} \mid \textsc{obl}_\theta\})_\sigma = (\uparrow_\sigma \textsc{arg}_1) \mid (\uparrow_\sigma \textsc{arg}_1)_{\sigma^{-1}} = \emptyset\}$

This in principle permits the s-structure feature $\textsc{arg}_1$, and thereby the agent of the event of writing, to be associated with either the grammatical function SUBJ, or $\textsc{obl}_\theta$ (e.g. in the passive), or indeed with no grammatical function (if, for example, the agent were unrealized syntactically). But the possibilities of complex predicates go beyond what is generally admitted for argument structure alternations in this (or any) argument structure model, at least with respect to simplex predicates. In the case of the complex predicate in (2), for example, the agent of the event of writing must be associated with the grammatical function $\textsc{obj}_\theta$, which is not possible in the Findlay/Asudeh *et al.* model.[21]

Whichever approach one takes to the representation of the meaning of predicates (e.g. whether along the lines of (22) or (23)), the

[21] The problem is that, as mentioned already, in traditional argument structure terms the agent of 'write' is '[−O]', but when embedded under the light verb it must be realized as '[+O]'.

solution to the problem at hand is in fact readily available in the glue system, and relatively simple to implement. The present exposition adopts the model of Asudeh and Giorgolo (2012).[22] Findlay's (2014) augmentations of Asudeh and Giorgolo's model are not crucial to the point at hand, so they are not utilized in this section, in order to simplify the discussion.

There is no need to change any of the basic assumptions regarding ordinary verbs like 'write'. That is, the lexical entry for 'write' will include the information in (23), just as under the proposals of Asudeh and Giorgolo (2012). As stated, the information in this lexical entry means that the f-structure SUBJ of a clause headed by 'write' will be associated with the $\textsc{arg}_1$ feature at s-structure, and thereby with the agent of the writing event in the meaning representation. Now let us assume that the Urdu light verb *de* 'let' has a lexical entry such as the following:[23]

(25) 'let' V
$(\uparrow \textsc{permissive}) = +$
$(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg}_1)$
$(\uparrow \textsc{obj}_\theta)_\sigma = (\uparrow_\sigma \textsc{arg}_3)$

$\lambda P.\lambda y.\lambda x.\lambda e.let(x, y, P(y, e)):$
$[(\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma] \multimap$
$(\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{arg}_3) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma$

$\lambda P.\lambda y.\lambda x.\lambda e.P(x, y, e):$
$[(\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{arg}_3) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma] \multimap$
$(\uparrow_\sigma \textsc{arg}_3) \multimap (\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma$

Consider first only the f-descriptions in the third and fourth lines and the first meaning constructor. According to the f-descriptions, the light verb requires that the f-structure for its clause contain both a SUBJ and an $\textsc{obj}_\theta$ argument, associated with the s-structure features $\textsc{arg}_1$ and $\textsc{arg}_3$ respectively. This essentially corresponds to the subcategorization for [−O] AGENT and [+R] GOAL argu-

[22] See (30) for the demonstration that the proposal would also work under a more standard treatment of glue expressions (i.e. using $(\uparrow \textsc{subj})_\sigma$ etc. rather than $(\uparrow_\sigma \textsc{arg}_1)$).

[23] I follow Butt (1998) in assuming that this verb does not introduce a new event variable, but the point is not crucial.

ments in (4) (merely with the argument structure mapping process resolved for the example under discussion). When combined with an ordinary transitive (or indeed intransitive) verb the specification $(\uparrow \textsc{subj})_\sigma \;=\; (\uparrow_\sigma \textsc{arg}_1)$ merely replicates that of the lexical verb, but the specification $(\uparrow \textsc{obj}_\theta)_\sigma \;=\; (\uparrow_\sigma \textsc{arg}_3)$ specification is new. The first meaning constructor also introduces a new entity variable into the meaning representation, referring to the permitter. By default, one would expect that if a word introduces a new grammatical function, and also introduces a new variable in the semantics corresponding to a grammatical function, then the meaning constructor introducing that variable will link it with the grammatical function via the semantic structure referred to in the corresponding glue term. This is what the first meaning constructor does: it associates the $\textsc{obj}_\theta$, via $\textsc{arg}_3$, with the 'permitter' role, and leaves the $\textsc{subj}$ function associated, via $\textsc{arg}_1$ with an argument of the embedded predicate. That is, if we combine the meaning of 'write' from (23) with the first meaning constructor in (25), we get:

(26)
$$\begin{array}{l}\lambda z.\lambda y.\lambda x.\lambda e.let(x,y,[write(e)\wedge agent(e,y)\wedge theme(e,z)]):\\ (\uparrow_\sigma \textsc{arg}_2) \multimap (\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{arg}_3) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma\end{array}$$

This may be what by default we should expect, but as it is, it does not work: the subject of the complex predicate should be the permitter, not the agent of the event permitted, and the $\textsc{obj}_\theta$ should be the agent of the event permitted, not the permitter. That is, the glue term $(\uparrow_\sigma \textsc{arg}_1)$ in (26) is linked with y, the agent of the event of writing, while the term $(\uparrow_\sigma \textsc{arg}_3)$ in (26) is associated with x, the 'permitter'. But since $\textsc{arg}_1$ is linked with $\textsc{subj}$, and $\textsc{arg}_3$ with $\textsc{obj}_\theta$, this means that, given the sentence in (2), with f-structure as in (19) = (27), Saddaf would be the permitter and Anjum the writer.

(27)
$$\left[\begin{array}{ll}\textsc{pred} & \text{'write'}\\ \textsc{permissive} & +\\ \textsc{subj} & \left[\begin{array}{ll}\textsc{pred} & \text{'Anjum'}\end{array}\right]\\ \textsc{obj}_{\text{goal}} & \left[\begin{array}{ll}\textsc{pred} & \text{'Saddaf'}\end{array}\right]\\ \textsc{obj} & \left[\begin{array}{ll}\textsc{pred} & \text{'note'}\end{array}\right]\end{array}\right]$$

This is where the second meaning constructor comes in. This makes no contribution to the meaning: it has an identity function on the meaning side. On the glue side, however, it takes an ordered set of glue premises and returns the same set in a different order. This reordering functions to effectively swap the associations between the glue terms $(\uparrow_\sigma \text{ ARG}_1)$ and $(\uparrow_\sigma \text{ ARG}_3)$ and the entity variables in the meaning representation, such that $(\uparrow_\sigma \text{ ARG}_1)$ is now linked with x, and $(\uparrow_\sigma \text{ ARG}_3)$ with y. ARG_1 is still linked with SUBJ, and ARG_3 with OBJ_θ, since these specifications cannot be changed, once made (in some sense, therefore, preserving the principle of Direct Syntactic Encoding). But we now have the correct associations between grammatical functions and thematic roles: SUBJ is linked to the permitter, and OBJ_θ to the writer. That is, if we compose the meaning constructor in (26) with the second meaning constructor in (25), the result is as shown in (28).

(28) $\lambda z.\lambda y.\lambda x.\lambda e.let(x, y, [write(e) \wedge agent(e, y) \wedge theme(e, z)])$:
$(\uparrow_\sigma \text{ ARG}_2) \multimap (\uparrow_\sigma \text{ ARG}_3) \multimap (\uparrow_\sigma \text{ ARG}_1) \multimap (\uparrow_\sigma \text{ EV}) \multimap \uparrow_\sigma$

The meaning constructor in (28) differs from that in (26) only in that the glue terms $(\uparrow_\sigma \text{ ARG}_1)$ and $(\uparrow_\sigma \text{ ARG}_3)$ are reordered. Crucially, this means that $(\uparrow_\sigma \text{ ARG}_1)$ is now associated with x on the meaning side, and $(\uparrow_\sigma \text{ ARG}_3)$ with y, rather than the other way around. x represents the 'permitter'; by the f-descriptions in the lexical entries for both 'write' and 'let', $(\uparrow_\sigma \text{ ARG}_1)$ is projected from $(\uparrow \text{ SUBJ})$; therefore, the SUBJ is now associated with the 'permitter', as it should be. Likewise, $(\uparrow_\sigma \text{ ARG}_3)$ is projected from $(\uparrow \text{ OBJ}_\theta)$, so this is associated with y, the agent of 'write'.

In this exposition I have treated the light verb 'let' as introducing two separate meaning constructors, but I do this purely for expository purposes: it is of course simpler to treat them as a single meaning constructor, which serves both to introduce the relevant meaning for the light verb, and to reorder the glue terms in such a way as to produce the correct associations between grammatical functions and semantic roles. That is, the lexical entry for 'let' given in (25) can be simplified by composing the two glue terms into one:

(29) 'let' V

$(\uparrow \textsc{permissive}) = +$

$(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg}_1)$

$(\uparrow \textsc{obj}_\theta)_\sigma = (\uparrow_\sigma \textsc{arg}_3)$

$$\lambda P.\lambda y.\lambda x.\lambda e.let(x, y, P(y, e)) :$$
$$[(\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma] \multimap$$
$$(\uparrow_\sigma \textsc{arg}_3) \multimap (\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma$$

According to the present proposal, the meaning constructor introduced by the light verb in the lexicon serves to control and constrain what is traditionally understood as 'argument fusion', in a rather more formally explicit way than is found in any other LFG literature. To summarize, the 'argument structure' associations between grammatical functions and s-structure $\textsc{arg}_x$ features, as specified in the lexical entries of lexical verbs by f-descriptions such as $(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg}_1)$, are not altered in any way by the light verb, because once they have been specified it is impossible to change them. But what the light verb can do is introduce new arguments, and new 'argument structure' associations between grammatical functions and s-structures features, and, crucially, it can reassociate the grammatical function – s-structure feature pairs with different semantic arguments in the meaning representation, which suffices to account for the usually rather mysterious process of 'argument fusion'.

The present proposal differs very clearly from the standard linking/XLE accounts, not only in its integration of a semantic representation, but also in its assumptions regarding the contribution of the light verb. Under the present proposal, the light verb does not introduce a new SUBJ argument, and does not cause the SUBJ of the embedded predicate to be demoted to $\textsc{obj}_\theta$, as in linking/XLE approaches.[24]

[24] According to the presentation in this section, the light verb does specify the existence of a SUBJ argument via the equation $(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg}_1)$ in the lexical entry, but as mentioned this is not a new contribution since it is already specified by the lexical verb. At least for the examples discussed in this paper, the fact that it is already specified by the lexical verb means that it is superfluous in the lexical entry for the light verb. It could perhaps, therefore, be removed, but I leave it in since there may potentially be contexts in which its presence is necessary.

Rather, it introduces a new OBJ_θ argument (= ARG_3), and then associates that OBJ_θ with the embedded agent.[25] At the same time it co-opts the SUBJ argument introduced by the lexical verb, and associates it with the new semantic role that its meaning introduces (i.e. in the example under discussion, the 'permitter').[26]

An empirically important difference between the present proposal and approaches that make use of the restriction operator is that under the present proposal the 'subject' of the embedded predicate, i.e. the permittee of permissive 'let' or the causee of a causative predicate, is not in fact a subject at any level of representation. This aligns with the Romance evidence discussed by Alsina (1996, 213–217) and Andrews (2007), where it is very clear that causees of causative complex predicates are not subjects, since only subjects can launch floating quantifiers, while causees are unable to do this. In a restriction-based approach the 'subject' of the embedded verb is still a subject at f-structure, merely not in the 'full' f-structure for the clause, so this constraint does not fall out so naturally.

[25] Or, more precisely, it associates it with the semantic role that is associated with SUBJ in the meaning constructor of the lexical verb, since this need not be an agent, of course.

[26] A more subtle difference between the present proposal and the linking approach, at least, is that the present proposal depends on the combinatory possibilities being stipulated in the lexical entries of the light verbs. For example, the additional argument introduced by 'let' in (29) is necessarily an OBJ_θ, such that this is the only possible grammatical function for the subject of the embedded predicate. As pointed out by an anonymous reviewer, in some respects the lack of formalization, and the resulting lack of constraints on argument fusion, in the linking approach could be considered advantageous; for example, Alsina and Joshi (1991) utilise the potential for variability in linking to account for differential case marking phenomena. In principle, of course, a fully formalized account with the same empirical coverage is to be preferred, and it does not seem in principle problematic to introduce optionality into the lexical entries of light verbs where necessary to simulate the variability that the linking approach affords. Further investigation is required to determine precisely what degree of freedom in linking is desired, and how well this could be formalized in the present approach. In this regard, a reviewer suggests that it may prove beneficial to introduce a more complex event structure representation into the semantics, e.g. as proposed by Butt and Ramchand (2005).

The proposal made here works just as well under a more traditional approach to verbal meaning constructors, i.e. that exemplified in (22). Under such an approach, the lexical entry for *de* 'let' would be:

(30) 'let' V

$(\uparrow \textsc{permissive}) = +$

$\lambda P.\lambda y.\lambda x.\lambda e.let(x, y, P(y, e)) :$
$[(\uparrow \textsc{subj})_\sigma \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma] \multimap$
$(\uparrow \textsc{obj}_\theta)_\sigma \multimap (\uparrow \textsc{subj})_\sigma \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma$

The meaning constructor in (30) introduces a new entity variable on the meaning side, representing the permitter, and by the order of the glue terms on the glue side this variable is associated with the semantic structure $(\uparrow \textsc{subj})_\sigma$. The variable that was associated with $(\uparrow \textsc{subj})_\sigma$ by the embedded predication becomes associated with $(\uparrow \textsc{obj}_\theta)_\sigma$. Note also that it would be trivial to rework this proposal within the 'First Order' glue of Kokkonidis (2008), or the propositional glue of Andrews (2010).

4 EXTENDING THE ANALYSIS

In this section, I show that the present proposal works unproblematically for the most complicated complex predicates treated in the linking/XLE literature, and in addition that it is able even to go beyond these approaches in dealing easily with phenomena that they cannot capture. I also discuss one formal drawback of the present proposal, which however does not affect the account of argument fusion and does not make the analysis any less adequate than the standard LFG analyses of other much less problematic phenomena.

To begin with, the present proposal has no difficulty in dealing with recursively embedded complex predicates, as found e.g. in Urdu. Butt *et al.* (2010) provide the following example of a nominal predicate quadruply embedded in a complex predicate, with the 'linking' style argument structure shown in (32); this is the most complex complex predicate I am aware of having been treated in the literature.[27]

[27] Following Butt (2014), I make a minor change to the ±O/R features in (32) compared with those assumed by Butt *et al.* (2010). The change is not crucial.

(31) *taaraa-ne* *amu-ko* *(bacce-se)*
Tara-ERG Amu-DAT child.OBL-INSTR
haathii *pinc* *kar-vaa* *le-ne*
elephant.M.SG.NOM pinch do-CAUS take-INF.OBL
dii-yaa
give-PERF.M.SG
‘Tara let Amu have the elephant pinched (by the child) (completely).’

(32)	‘let’ ⟨	AG	GO	‘take’ ⟨	AG	CAUS ⟨	AG	PAT	‘do’ ⟨	AG	‘pinch’ ⟨	AG	TH ⟩⟩⟩⟩⟩
		[−O]	[+O]							([−O])			[−R]
		|	|							|			|
		SUBJ	OBJ$_\theta$							OBL$_\theta$			OBJ

The core element of this verb form is a Noun-Verb complex predicate consisting of the predicate noun *pinc* ‘pinch’, and the light verb *kar* ‘do’. This is embedded under a causative predicate, which is realized morphologically on the light verb *kar* (but which has scope over the whole Noun-Verb predicate). This is further embedded under the ‘completive’ aspectual light verb *le* (the lexical meaning of which is ‘take’). Finally, this four-part predicate is embedded under the permissive light verb *de* ‘let’, which we saw in (2). I assume the following lexical entries for the verb forms and morphemes involved, with the permissive unchanged from (29).[28]

(33) ‘pinch’ N
$(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg}_1)$
$(\uparrow \textsc{obj})_\sigma = (\uparrow_\sigma \textsc{arg}_2)$

$\lambda y.\lambda x.\lambda e.pinch(e) \wedge agent(e,x) \wedge patient(e,y)$:
$(\uparrow_\sigma \textsc{arg}_2) \multimap (\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma$

[28] It is not particularly important for the present purposes precisely how one divides the meaning of the Noun-Verb complex predicate ‘do a pinch’ between the N and the V, i.e. between (33) and (34). The analysis assumed here associates the whole meaning of the Noun-Verb complex with the noun, which corresponds most closely with what Butt *et al.* (2010) assume in their linking-based presentation. The XLE analysis would be somewhat different, however, with ‘pinch’ introducing only an object(/patient) argument, and the subject(/agent) argument being introduced only by the light verb *kar* ‘do’.

(34) ‘do’ V

$(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg}_1)$

$\lambda P.\lambda x.\lambda e.P(x,e):$
$[(\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma] \multimap$
$(\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma$

(35) CAUSE

$(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg}_1)$
$(\uparrow \textsc{obl}_\theta)_\sigma = (\uparrow_\sigma \textsc{arg}_4)$
$(\uparrow \textsc{cause}) = +$

$\lambda P.\lambda y.\lambda x.\lambda e.cause(x,y,P(y,e)):$
$[(\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma] \multimap$
$(\uparrow_\sigma \textsc{arg}_4) \multimap (\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma$

(36) ‘take’ V

$(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg}_1)$
$(\uparrow \textsc{completive}) = +$

$\lambda P.\lambda x.\lambda e.completely(P(x,e)):$
$[(\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma] \multimap$
$(\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma$

(37) ‘let’ V

$(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg}_1)$
$(\uparrow \textsc{obj})_\sigma = (\uparrow_\sigma \textsc{arg}_3)$
$(\uparrow \textsc{permissive}) = +$

$\lambda P.\lambda y.\lambda x.\lambda e.let(x,y,P(y,e)):$
$[(\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma] \multimap$
$(\uparrow_\sigma \textsc{arg}_3) \multimap (\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma$

Essentially, the causative predicate associates $\textsc{arg}_4$ with the agent of the pinching, and reassociates $\textsc{arg}_1$ with the causer. The permissive reassociates the causer with $\textsc{arg}_3$, and $\textsc{arg}_1$ with the permitter. Composing all the relevant meanings together will produce the meaning constructor in (38); the glue proof for this derivation is shown in Figure 1 on p. 454.

(38) $\lambda z.\lambda y.\lambda x.\lambda w.\lambda e.let(w,x,completely(cause(x,y,(pinch(e) \wedge agent(e,y) \wedge patient(e,z))))) : (\uparrow_\sigma \textsc{arg}_2) \multimap (\uparrow_\sigma \textsc{arg}_4) \multimap (\uparrow_\sigma \textsc{arg}_3) \multimap (\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma$

The s-structure feature ARG$_1$ is linked to SUBJ, meaning that the SUBJ is understood as the permitter; ARG$_2$ is linked to OBJ, meaning that OBJ is understood as the patient of the pinching event; ARG$_3$ is linked to OBJ$_\theta$, meaning that OBJ$_\theta$ is understood as the causer of the pinching event; ARG$_4$ is linked to OBL$_\theta$, meaning that OBL$_\theta$ is understood as the agent of the pinching event. So, the f-structure for (31) will be as in (39) which, in association with the meaning constructor in (38), will result in the correct interpretation.

(39)

$$\begin{bmatrix} \textsc{pred} & \text{‘pinch’} \\ \textsc{cause} & + \\ \textsc{permissive} & + \\ \textsc{completive} & + \\ \textsc{subj} & \begin{bmatrix} \textsc{pred} & \text{‘Tara’} \end{bmatrix} \\ \textsc{obj} & \begin{bmatrix} \textsc{pred} & \text{‘elephant’} \end{bmatrix} \\ \textsc{obj}_\theta & \begin{bmatrix} \textsc{pred} & \text{‘Amu’} \end{bmatrix} \\ \textsc{obl}_\theta & \begin{bmatrix} \textsc{pred} & \text{‘child’} \end{bmatrix} \end{bmatrix}$$

The present proposal is thus able to deal with even very complex complex predicates; there is no reason why it should not be able to deal with essentially the same range of phenomena that can be dealt with under the linking and XLE approaches. There is, however, one respect in which the present proposal may be at a disadvantage relative to the traditional approaches, and which is relevant to the analysis of (31). The monoclausality assumed for the f-structure in §3 has one unfortunate consequence: there are no necessary constraints on the order of composition of predicates. That is, while the desired meaning (in (38)) can be correctly derived from the premises (as shown in Figure 1), it is also possible to derive a number of incorrect interpretations, by applying the light verbs' meaning constructors in different orders. Essentially, this is a problem of scope. The advantage of the multiclausal analysis obtained by using the restriction operator is that the order of embedding of the predicates can be constrained. The XLE analysis of Urdu complex predicates implies that the order of embedding must reflect the tree structure (though at least with unembedded complex predicates, there is no necessary subordination of the matrix verb under the light verb, or vice versa, in c-structure terms). The same is true of Romance complex predicates, as discussed by Alsina (1997) and e.g. Andrews (2007). That is, for example, each recursively embedded complex predicate will form a subconstituent of the larger verbal constituent. Under the linking/XLE approaches, this will result in an f-structure semantic form that shows the correct embedding. Under the present proposal there is no embedding in semantic forms, and there is no immediately available means of enforcing the correct order of embedding in the semantics. However, there are two main reasons why this apparent disadvantage of the present approach is not fatal.

To begin with, although the linking/XLE approaches are capable of obtaining the correct order of embedding in the f-structure semantic form, it is not at all obvious that they could easily achieve the same in a glue-based semantic representation, if they were augmented with such. That is, the problem with the present proposal is no more a problem than it is for the traditional accounts, if only the semantics is considered (and part of the present proposal is that only the semantics is relevant, since there is no predicate composition in f-structure). Any proposal that assumes a monoclausal f-structure (such as Butt 1995 and Alsina 1996) would be unable to account for the order of composition in glue. A restriction-based account seems less problematic, because there are distinct f-structures for each level of embedding, but restriction leaves these distinct f-structures essentially dissociated. This means that there would be no easy way for the meaning constructor of the light verb to refer to the (s-structure projected from the) f-structure associated with the predicate embedded under it.[29] The only way to constrain the glue composition effectively by reference to f-structure is to assume an embedded f-structural representation, as proposed by Andrews and Manning (1999) and as exemplified in (14). However, no standard LFG analysis assumes this, and as discussed above it rather undermines the basic intuition of monoclausality.

Secondly, the difficulty with constraining semantic scope when the f-structure is flat is not unique to complex predicates. As discussed

[29] So it is not clear that a restriction-based account is even compatible with a glue-based semantic analysis. The problem may possibly be resolvable if the f-descriptions in the c-structure specified that the s-structures projected from the dissociated f-structures be embedded one inside the other, but the details of this remain to be explored.

e.g. by Andrews and Manning (1999), it is a long-term problem in the analysis of recursive modification. Recursive modification involving one or more intensional adjectives must be interpreted semantically with respect to the linear / hierarchical order, e.g.:

(40) a. The former trustworthy chairman.
b. The trustworthy former chairman.

In LFG, the ADJUNCT set in which such modifiers appear at f-structure is flat, such that there is no way for the interpretative constraint to be enforced in the semantics. This is already a problem for LFG, then, and whatever solution may be proposed can be easily extended to the analysis of complex predicates, such that this should not be considered a fatal flaw of the present proposal.[30]

This difficulty aside, there is one important respect in which the present proposal is descriptively superior to the linking and XLE approaches. Butt *et al.* (2010) note that the OBL_θ in sentences like (31) is optional, and should perhaps be treated as an adjunct, but that this is not done in their linking analysis because "argument suppression with respect to argument merger as part of complex predication is not predicted within Linking Theory." That is, the linking approach to complex predicates has no way to deal with the optionality of arguments. This is also impossible within XLE, since there is no way to remove an argument from the subcategorization list of a predicate.

[30] Besides the proposal of Andrews and Manning (1999), another proposed solution is under development by Andrews (2015). Both of these involve rather severe changes to the traditional LFG view of f-structure. Note that neither f-precedence, nor the notion of linear precedence discussed by Asudeh (2009), can handle the complex predicate data, since the crucial relation is c-structure hierarchy, and not necessarily linearity. In XLE, surface scope and surface adjunct scope can be captured at f-structure using the notations <s / >s and ∈<h<s / ∈<h>s respectively, but it is not immediately obvious how this information could be utilized formally to constrain a glue derivation. Perhaps the simplest alternative is simply to state a constraint to the effect that semantic composition should mirror the c-structure, equivalent to the constraint placed on predicate composition by Alsina (1997, 237–238), though it must be admitted that such a solution is rather informal, and not easily formalized.

Under the present proposal, optionality of arguments would be unproblematic. Butt *et al.* (2010) suggest a possible adjunct analysis for the optional element, presumably because this is the default interpretation for an optional phrase. However, work by Needham and Toivonen (2011), Christie (2013), and Toivonen (2013) show that the argument-adjunct distinction is not absolute, and that optionality may also be a feature of some arguments. At least for the present purposes, given that the standard approaches to complex predicates in LFG assume that the element in question is an argument, an analysis as an optional argument seems preferable to an analysis as an adjunct. Asudeh and Giorgolo (2012) and Asudeh *et al.* (2014) formalize a semantics-based account of optional arguments of simplex predicates, and this can easily be transferred to the present analysis. Specifically, the optional argument is the causee of the (morphological) CAUSE predicate. Cf. the following example, based on the relevant portion of the complex predicate in (31).

(41)	*amu-ne*	*(bacce-se)*	*haathii*	*pinc*
	Amu-ERG	child.OBL-INSTR	elephant.M.SG.NOM	pinch
	kar-vaa-yaa			
	do-CAUS-PERF.M.SG			

‘Amu had the elephant pinched (by the child).’

Instead of the lexical contribution in (35) for the causative element, we can assume the contribution in (42). I slightly update Asudeh and Giorgolo's representations based on Findlay (2014) and Asudeh *et al.* (2014), but treat the variability in grammatical function assignment as already resolved, since it is not relevant to the point at hand and would only complicate the discussion.[31]

[31] That is, in the first line of (42) I assume $\{(\uparrow \text{SUBJ})_\sigma = (\uparrow_\sigma \text{ARG}_1) \mid (\uparrow_\sigma \text{ARG}_1)_{\sigma^{-1}} = \emptyset\}$ rather than Asudeh *et al.*'s $\{(\uparrow \{ \text{SUBJ} \mid \text{OBL}_\theta \})_\sigma = (\uparrow_\sigma \text{ARG}_1) \mid (\uparrow_\sigma \text{ARG}_1)_{\sigma^{-1}} = \emptyset\}$, and make the equivalent simplification in the second line. Since processes such as passivization, etc., are not at issue here, the option of either SUBJ or OBL_θ in the first line will necessarily resolve to SUBJ, and the same option in the second line will necessarily resolve to OBL_θ, in accordance with Kibort's (2007) Mapping Principle, so it is simpler here to ignore the optionality.

(42) $\textsc{cause}$

$$\{(\uparrow \textsc{subj})_\sigma \;=\; (\uparrow_\sigma \textsc{arg}_1) \mid (\uparrow_\sigma \textsc{arg}_1)_{\sigma^{-1}} \;=\; \emptyset\}$$
$$\{(\uparrow \textsc{obl}_\theta)_\sigma \;=\; (\uparrow_\sigma \textsc{arg}_4) \mid (\uparrow_\sigma \textsc{arg}_4)_{\sigma^{-1}} \;=\; \emptyset\}$$

$$\begin{array}{l} \lambda P.\lambda y.\lambda x.\lambda e.cause(x,y,P(y,e)) : \\ [(\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma] \multimap \\ (\uparrow_\sigma \textsc{arg}_4) \multimap (\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma \end{array}$$

$$(\lambda P.\exists x.P(x) : ((\uparrow_\sigma \textsc{arg}_4) \multimap \uparrow_\sigma) \multimap \uparrow_\sigma)$$

The f-descriptions in the first two lines of the lexical entry introduce the two arguments of the CAUSE predicate. The first line states that either there will be an f-structure SUBJ which projects to the semantic structure $\textsc{arg}_1$, or else there is nothing in the f-structure which projects to $\textsc{arg}_1$. Likewise, the second line states that either there will be an f-structure $\textsc{obl}_\theta$ which projects to the semantic structure $\textsc{arg}_4$, or else there is nothing in the f-structure which projects to $\textsc{arg}_4$. In the present context there is nothing to license the absence of a SUBJ from the f-structure. However, the rest of the lexical entry does license the absence of the $\textsc{obl}_\theta$ argument from the f-structure. The first meaning constructor is unchanged from (35): it introduces a new entity variable, the 'causer', and rearranges the associations between s-structure $\textsc{arg}_x$ features and variables, such that $\textsc{arg}_1$ will be associated with the causer and $\textsc{arg}_4$ with the causee.

The crucial element is the second meaning constructor in (42). This optional meaning constructor existentially quantifies the variable associated with $\textsc{arg}_4$. If, then, the $\textsc{obl}_\theta$ argument is absent from the f-structure, i.e. if no causee is explicitly realized in the syntax, this meaning constructor can apply to quantify the variable that would otherwise be left hanging. If the causee is explicitly realized in the syntax, appearing as $\textsc{obl}_\theta$ at f-structure, then this will serve to quantify the variable associated with $\textsc{arg}_4$, and the optional meaning constructor in (42) will not be required. That is, there are two possible f-structures for the example in (41), depending on whether or not the causee is omitted:

(43)

$$\begin{bmatrix} \textsc{pred} & \text{'pinch'} \\ \textsc{cause} & + \\ \textsc{subj} & \begin{bmatrix} \textsc{pred} & \text{'Amu'} \end{bmatrix} \\ \textsc{obj} & \begin{bmatrix} \textsc{pred} & \text{'elephant'} \end{bmatrix} \\ \textsc{obl}_\theta & \begin{bmatrix} \textsc{pred} & \text{'child'} \end{bmatrix} \end{bmatrix}$$

(44)
$$\begin{bmatrix} \textsc{pred} & \text{‘pinch’} \\ \textsc{cause} & + \\ \textsc{subj} & \begin{bmatrix} \textsc{pred} & \text{‘Amu’} \end{bmatrix} \\ \textsc{obj} & \begin{bmatrix} \textsc{pred} & \text{‘elephant’} \end{bmatrix} \end{bmatrix}$$

Assuming the simplified noun meanings in (45), and assuming the simplified ‘finiteness’ meaning constructor in (46) to quantify the event variable, the resulting meaning constructors for (43) and (44) will be as in (47) and (48) respectively. The glue proofs for these derivations appear in Figures 2 and 3 respectively, on pp. 455 and 456.

(45) a. $Amu : \uparrow_\sigma$

b. $elephant : \uparrow_\sigma$

c. $child : \uparrow_\sigma$

(46) $\lambda P.\exists e.P(e) : ((\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma) \multimap \uparrow_\sigma$

(47) $\exists e.cause(Amu, child, (pinch(e) \wedge agent(e, child) \wedge$
$patient(e, elephant))) : \uparrow_\sigma$

(48) $\exists e.\exists y.cause(Amu, y, (pinch(e) \wedge agent(e, y) \wedge$
$patient(e, elephant))) : \uparrow_\sigma$

In this way, the present proposal for dealing with complex predicates can easily handle the optionality of arguments, in a way that neither the linking approach nor the XLE approach can.

5 COMPARISON WITH PREVIOUS GLUE APPROACHES

There exist a few previous treatments of complex predicate formation within LFG that make significant reference to semantics, though there are none within the current standard ‘new’ glue approach, and none that have been widely adopted. In this section, I briefly discuss each approach, and provide comparisons with my own proposals.

The earliest proposal was made by Kaplan and Wedekind (1993). They do not explicitly make use of glue, but Dalrymple *et al.* (1993a) briefly illustrate how their proposal would be represented in glue. As mentioned in §2, Kaplan and Wedekind (1993) introduced the restriction operator into LFG, and into the analysis of complex predicates.

They assume that the lexical entry of a verb like Urdu *likh* 'write' contains the following default specifications:

(49) a. $(\sigma \uparrow \text{ARG1}) = \sigma(\uparrow \text{SUBJ})$
b. $(\sigma \uparrow \text{ARG2}) = \sigma(\uparrow \text{OBJ})$

Difference of notation aside, this is identical to the third and fourth lines of (23). Kaplan and Wedekind (1993) further assume that a lexical redundancy rule exists that can systematically modify these specifications for any ordinary verb, such that they become:

(50) a. $(\sigma\ [\uparrow \backslash\text{SUBJ}]\ \text{ARG1}) = \sigma(\uparrow \text{OBJ2})$
b. $(\sigma\ [\uparrow \backslash\text{SUBJ}]\ \text{ARG2}) = \sigma(\uparrow \text{OBJ})$

As described by Dalrymple *et al.* (1993a, 16), this means that the meaning for *likh* 'write' will be as in (51) (using the original glue notation). This will combine with the meaning for the permissive light verb, *de* 'let', which is shown in (52). The 'new glue' (Dalrymple *et al.* 1999) versions of (51) and (52) are shown in (53) and (54) respectively.

(51) $(\uparrow \backslash\text{SUBJ})_\sigma \rightsquigarrow \textit{write}(X, Y)$
where X is the meaning of the OBJ2, and Y is the meaning of the OBJ.

(52) $\uparrow_\sigma \rightsquigarrow \textit{permit}(X, Y)$
where X is the meaning of the SUBJ, and Y is the meaning of $\uparrow \backslash$SUBJ.

(53) $\lambda y.\lambda x.\textit{write}(x, y) : (\uparrow \text{OBJ})_\sigma \multimap (\uparrow \text{OBJ}_\theta)_\sigma \multimap (\uparrow \backslash\text{SUBJ})_\sigma$

(54) $\lambda P.\lambda x.\textit{permit}(x, P) : (\uparrow \backslash\text{SUBJ})_\sigma \multimap (\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma$

As discussed in §2, it may be preferable to avoid the use of the restriction operator in any case, but this is particularly true when one starts using it to refer to semantic structures projected from f-structures. But the most serious problem with the proposal of Kaplan and Wedekind (1993), which was noted by e.g. Dalrymple *et al.* (1993a), Butt (1994), Andrews and Manning (1999), and Butt *et al.* (2003), is that it assumes a fundamentally lexical approach to complex predicate formation and argument fusion. Specifically, the operation that serves to reassign the ARG_1 of 'write' to OBJ2 $(=\text{OBJ}_\theta)$

applies in the lexicon. As pointed out by Dalrymple *et al.* (1993a, 16), Kaplan and Wedekind's proposal predicts that any ordinary lexical verb can combine with only a finite number of light verbs, and entails a considerable amount of lexical duplication: there must exist separate lexical entries for a verb that combines with one light verb, with two light verbs, etc., and for light verbs that appear as the only light verb in a sentence, or with one other light verb in the sentence, etc. To the extent that Kaplan and Wedekind's semantic proposals can be converted to apply within the framework of Butt *et al.* (2003), who show that the restriction operator can be used to permit predicate composition in the syntax, they would unavoidably be affected by the problems with the linking and (especially) XLE approaches described in §2.

An alternative proposal is made by Dalrymple *et al.* (1993a), followed by Zaenen and Dalrymple (1995, 1996). Their proposals are formalized in the original glue representation.[32] Their proposal is that the links, or mapping, between the syntactic arguments and semantic roles of verbs are not defined in the lexical entries of those verbs, but are derived from independent 'mapping rules', which are universally available in the analysis of any clause. For example, they propose the following mapping rule (p. 8), which can apply to any clause containing a simple transitive verb selecting for an agent and a theme argument:

(55) $$!(\forall f,X,Y.((f\ \textsc{subj})_\sigma \rightsquigarrow X) \otimes ((f\ \textsc{obj})_\sigma \rightsquigarrow Y) \multimap \\ \mathit{agent}((f\ \textsc{pred})_\sigma,X) \otimes \mathit{theme}((f\ \textsc{pred})_\sigma,Y))$$

They explain this rule as follows:

> This rule associates subjects with agents, and objects with themes. It states that for all f-structures f, if the SUBJ of f is X and the OBJ of f is Y, we can conclude that X is the f-structure's PRED's agent, and Y is the f-structure's PRED's theme. (p. 8)

[32] The original glue representation was introduced by Dalrymple *et al.* (1993b); Dalrymple *et al.* (1996) replaced this with a formally simpler system, the first to gain wide currency; the current 'new' glue representation was introduced by Dalrymple *et al.* (1999).

When it comes to complex predicates, there is no alteration or manipulation of the mappings between grammatical functions and semantic roles (since these are not defined in the lexicon). A lexical verb introduces one or more grammatical functions and one or more semantic roles, and a light verb can also introduce a grammatical function and a semantic role. Then the correct mapping rule is selected that can match up all the pairs in the clausal f-structure, both those introduced by the lexical verb and those introduced by the light verb. For example, Dalrymple *et al.* propose the following mapping rule for a sentence with a permissive light verb and a lexical verb with agent and theme:

(56) $!(\forall f,X,Y,Z.((f\ \textsc{subj})_\sigma \rightsquigarrow X) \otimes ((f\ \textsc{obj})_\sigma \rightsquigarrow$
$Y) \otimes ((f\ \textsc{obj2})_\sigma \rightsquigarrow Z) \multimap \mathit{permitter}((f\ \textsc{pred})_\sigma, X) \otimes$
$\mathit{agent}((f\ \textsc{pred})_\sigma, Z) \otimes \mathit{theme}((f\ \textsc{pred})_\sigma, Y))$

It is not possible to directly convert this proposal into the 'new' glue representation, because the mapping rules proposed mix the meaning language and linear implication in a way that is no longer possible. However, the spirit of the proposal can be implemented. Authors such as Asudeh *et al.* (2008, 2013, 2014), Haug (2008) and Lowe (2015) assume that the meaning of verbs can be broken down into a basic verbal meaning and a semantic role or argument structure template. So, for the Urdu verb *likh* 'write', in place of the lexical entry with a single meaning constructor (23), we can assume a lexical entry such as the following:

(57) 'write' V
$(\uparrow \textsc{pred}) = \text{'write'}$
$\lambda e.\mathit{write}(e) : (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma$

$(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg}_1)$
$(\uparrow \textsc{obj})_\sigma = (\uparrow_\sigma \textsc{arg}_2)$
$\lambda P.\lambda y.\lambda x.\lambda e.P(e) \wedge \mathit{agent}(e,x) \wedge \mathit{theme}(e,y) :$
$[(\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma] \multimap$
$(\uparrow_\sigma \textsc{arg}_2) \multimap (\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma$

The advantage of this is that it raises the possibility of generalizing over both the syntactic and semantic aspects of argument structure patterns using templates (Dalrymple *et al.* 2004; Asudeh *et al.* 2008, 2013). For the present purposes, however, the relevant point is that the second meaning constructor in (57) contains the associations between semantic roles and grammatical functions (via s-structure features), which is the component of sentential meaning that Dalrymple *et al.* (1993a) propose is not a part of lexical entries, but universally available. So, if we were to convert Dalrymple *et al.*'s proposal into a format that conforms with the proposals of Asudeh and Giorgolo (2012) in 'new' glue, we would require the following lexical entry for *likh* 'write':

(58) 'write' V
$(\uparrow \textsc{pred}) = \text{`write'}$
$\lambda e.write(e) : (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma$
$(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg}_1)$
$(\uparrow \textsc{obj})_\sigma = (\uparrow_\sigma \textsc{arg}_2)$

This requires that the verb appear in an f-structure with a SUBJ and an OBJ, and also requires that the SUBJ and the OBJ project s-structures $\textsc{arg}_1$ and $\textsc{arg}_2$ respectively. But it makes no statement about how those grammatical functions, or those s-structure features, relate to the semantically entailed participants of the event of writing. This proposal would then require that the meaning constructor in (59) be universally available in the analysis of any sentence, and that in the analysis of a sentence containing the verb 'write' it be used to provide the appropriate semantic relations for the verb based on the $\textsc{arg}_x$ features specified in the verb's lexical entry.[33]

(59) $\lambda P.\lambda y.\lambda x.\lambda e.P(e) \wedge agent(e,x) \wedge theme(e,y) : [(\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma] \multimap (\uparrow_\sigma \textsc{arg}_2) \multimap (\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma$

The universally available meaning constructor that would be required in the case of a complex predicate such as that in (2) would be as follows (i.e. in place of (56)):

(60) $\lambda P.\lambda Q.\lambda z.\lambda y.\lambda x.P(x,y,[Q(e) \wedge agent(e,y) \wedge theme(e,z)]) : [(\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma] \multimap (\uparrow_\sigma \textsc{arg}_2) \multimap (\uparrow_\sigma \textsc{arg}_4) \multimap (\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma$

[33] I.e. (59) replaces (55) in the original formulation.

The meaning constructor in the lexical entry of the permissive light verb *de* 'let' would then be:

(61) $\lambda P.\lambda e.\mathit{permit}(P(e)) : [(\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma] \multimap (\uparrow_\sigma \textsc{ev}) \multimap \uparrow_\sigma$

This is slightly different from the original proposal of Dalrymple *et al.* (1993a), since for them verbs do contain specification of their thematic roles in the lexicon, and it is merely the links between those roles and grammatical functions that are specified by the mapping rules. In new glue these cannot be separated without entirely losing the link between semantic role and grammatical function. A full separation would be possible within the proposals of Lowe (2014), where the use of complex typed structures permits meaning constructors to be effectively partitioned in two, but the resulting analysis for complex predicates would be further from Dalrymple *et al.*'s original proposals than the suggestion just made. So, in Lowe's (2014) model, the meaning constructor for a verb like 'write' would be as in (62), while the meaning constructor that would be removed from the lexicon and made universally available would be that in (63).

(62) $\lambda y.\lambda x.\lambda e.\mathit{write}(e) \wedge \mathit{agent}(e,x) \wedge \mathit{theme}(e,y) : (\uparrow_\sigma \textsc{rel})_{\langle e \rightarrow e \rightarrow e \rightarrow t\rangle}$

(63) $\lambda P.\lambda y.\lambda x.\lambda e.P(x,y,e) : (\uparrow_\sigma \textsc{rel})_{\langle e \rightarrow e \rightarrow e \rightarrow t\rangle} \multimap (\uparrow_\sigma \textsc{arg}_2)_{\langle e\rangle} \multimap$
$(\uparrow_\sigma \textsc{arg}_1)_{\langle e\rangle} \multimap (\uparrow_\sigma \textsc{ev})_{\langle e\rangle} \multimap \uparrow_{\sigma\langle e\rangle}$

Whether in its original form, or in one way or another converted to new glue, perhaps the main disadvantage of Dalrymple *et al.*'s (1993a) proposal is that it requires a potentially large inventory of universally available meaning constructors to function as mapping tools, all of which are available in any one derivation.[34] So the first mapping rule discussed above (55) associates subjects with agents and objects with themes, but there must also be different meaning constructors for every potential combination of grammatical functions and thematic roles, including for complex predicates, and in principle all are available for any sentence (though only the correct one will work, of course).

The mapping rules that Dalrymple *et al.* (1993a, 18) propose are of a rather different nature from other meaning constructors: "Map-

[34] Besides the comments here, compare also the comments and criticism on Dalrymple *et al.*'s proposals by Andrews and Manning (1999, 136–141).

ping rules exist separate from the collections of formulas that contain meanings of sentences." That is, it would be necessary to assume something additional in the grammar, alongside the standardly assumed structures and projections, specifically in order to deal with complex predicates. In fact, the ability to generalize mapping possibilities across verbs is readily available by making use of templates to encode generalizations across lexical entries, as shown by Asudeh and Giorgolo (2012) and Asudeh *et al.* (2014).

A further problematic aspect of Dalrymple *et al.*'s proposal is that, at least in the original formulation, these mapping rules necessarily make use of the 'of course' operator !, since each one can be used zero or more times in any derivation. Asudeh and Crouch (2002, 28) and Asudeh (2012, 101) argue that ! can and should be kept out of the linear logic fragment used in glue, in order to protect the resource sensitivity of glue semantics. Whatever the formulation, it remains the case that the mapping rules or meaning constructors concerned must be allowed to apply as many times as necessary in any derivation, weakening the resource sensitivity of the semantic model.

Having said all that, the proposal of Dalrymple *et al.* (1993a) does appear to work: it is a fully formalized semantically integrated account of complex predicate formation that does not rely on manipulable PRED features and predicate composition in the f-structure, and that does not depend on a nebulous concept of argument fusion. These features are precisely what the present proposal aspires to.

Another early proposal for a semantic analysis of complex predicates was made by Andrews and Manning (1999, 119–128). Their proposal depends on a somewhat non-standard syntactic analysis of complex predicates, and the approach in general has not been widely adopted, even by the authors themselves; I will not therefore discuss the proposals of Andrews and Manning (1999, 119–128) here, but focus on the more recent proposal of Andrews (2007). Andrews' (2007) proposal for a semantic analysis of complex predication is in some respects the most similar existing account to the present proposal, but it is formalized in a non-standard approach to glue, and to the LFG projection architecture, developed by Andrews (2010). Like Dalrymple *et al.* (1993a) and the present proposal, Andrews (2007) is concerned with the question of argument fusion, and proposes the following meaning constructor for a causative light verb predicate:

(64) $\lambda P.\lambda y.\lambda x.Cause(x, y, P(y)) : ((\uparrow \text{?OBJ})_e \rightarrow \uparrow_p) \rightarrow (\uparrow \text{?OBJ})_e \rightarrow (\uparrow \text{SUBJ})_e \rightarrow \uparrow_p$

where $(\uparrow \text{SUBJ})_e$ and $\uparrow_p$ correspond to $(\uparrow \text{SUBJ})_\sigma$ and $\uparrow_\sigma$ respectively in the more standard approach to the LFG architecture assumed here. $(\uparrow \text{?OBJ})_e$ is essentially a place-holder for a more sophisticated statement governing grammatical function alternations, since in the Romance phenomena that Andrews addresses, the causee may surface as either a dative case OBJ_θ or an accusative case OBJ, depending on whether the embedded predicate is transitive or intransitive, respectively.[35] Andrews (2007) does suggest how a more sophisticated statement might be formulated, but the presentation is brief and the proposal is not explained or exemplified in full. Altogether, the proposal is hard to assess for this reason; it seems to be heading in a similar direction to the present proposal, but the presentation is elliptical and, as stated, it is formalized in a non-standard approach to semantics in LFG.

The most recent proposal regarding complex predicates in LFG is made by Homola and Coler (2013). This proposal is in certain respects reminiscent of that of Dalrymple *et al.* (1993a), but it is formally rather different. Homola and Coler (2013) propose a radically new approach to the syntax-semantics interface in LFG, the details of which are beyond the scope of the present discussion. They deal firstly with the question of predicate composition, proposing a means of permitting predicate composition in the f-structure without having to make use of the restriction operator. Their proposal in this respect is essentially parallel to Dalrymple *et al.*'s (1993a) proposal, but focused on the f-structure rather than semantics. They propose to use *equational unification*, a concept from logical programming, to model predicate composition in f-structure. A set of 'equational theories' E_i constitute a separate subcomponent of the grammar. Homola and Coler propose the semantic forms in (65), and the equational theory in (66), to model the predicate fusion of a causative predicate with an intransitive verb:

(65) a. $\text{CAUSE}\langle (\uparrow \text{SUBJ}), \text{f}\langle (\uparrow \text{OBJ})\rangle\rangle$
b. $\text{laugh}\langle (\uparrow \text{SUBJ})\rangle$

[35] In the present model, this alternation should fall out unproblematically with the addition of the Findlay-Asudeh *et al.* (2014) argument structure proposals, depending on precisely how the Mapping Principle is formulated.

(66) $E = \{\textsc{cause}\langle\ (\uparrow \textsc{subj}), \mathrm{f}\langle\ (\uparrow \textsc{obj})\rangle\rangle \approx \mathrm{f}\langle\ (\uparrow \textsc{subj})\rangle\}$'

The equational theory in (66) functions to produce the complex semantic form in (67) from those in (65).

(67) CAUSE⟨ (↑ SUBJ), laugh⟨ (↑ OBJ)⟩⟩

It is evident that this works according to essentially the same principle as the proposal of Dalrymple *et al.* (1993a): a separate component of the grammar contains a set of formulae that specify the argument reassignment/fusion in complex predicates. It therefore suffers from the same problem. A large number of such formulae must be assumed to deal with the full variety of complex predicates and all may be available in any derivation. As for the semantics, Homola and Coler (2013) need a special formula to appear on the c-structure node dominating a lexical verb which, in relation to the causative example they discuss, permits either the SUBJ or the OBJ to function as the actor of the lexical verb (since by their defaults, the subcategorization of the lexical verb would require the SUBJ to fill this role).[36] In this case too, one would presumably need a whole set of different formulae, any of which could potentially apply in any given case. For example, a formula would be required that enabled the $\textsc{obj}_\theta$ to fill the actor role, to cover complex predicates such as the permissive with a transitive predicate. All in all, their proposal involves a thorough revision of the LFG architecture, the implications of which would have to be carefully analysed, yet from the present perspective it still suffers from some of the same problems that already affected earlier proposals made within a more standard model.

While there have been a number of earlier proposals for a semantically integrated account of predicate composition in LFG, and while one or two of these at least show the potential to provide a descriptively adequate account of complex predication (Dalrymple *et al.* 1993a; Andrews 2007), none have been developed in great detail beyond the initial proposal, none have been adopted more widely in the LFG community, and none are formulated (or could easily be reformulated) in the 'new' glue approach, which has been standard in LFG for

[36] It is not worth providing the formulae they propose, since they could only be understood in the wider context of their proposals regarding the syntax-semantics interface in LFG, which as stated is beyond the scope of this paper.

over fifteen years. If in no other respect, then, the present proposal advances on previous proposals simply because it is formulated within the standard approach to LFG + glue, and therefore its potential for wider adoption is correspondingly greater.

6 CONCLUSION

In this paper, I have proposed a new, semantically integrated account of complex predicate formation within LFG + glue. I have shown that the proposed approach to complex predicates can deal with all the data that the standard linking/XLE approaches can deal with, even recursive complex predicate structures. Moreover, the proposed approach improves upon the standard linking/XLE approaches because it is fully formalized (in contrast to the linking approach, at least), does not involve mysterious processes of 'predicate composition' and 'argument fusion', does not require the use of construction-specific mechanisms (such as the restriction operator, manipulable PREDs, etc.), and properly integrates glue semantics. Previous accounts of complex predicates in LFG that integrate semantics either suffer from certain problems, or are not fully developed, but the present proposal is fully formalized within recent approaches to argument structure in LFG + glue, shares none of the problems affecting previous proposals, and involves no construction-specific additions to the formal model.

The one apparent weakness of the proposal, relating to the scope of multiple light verbs in a doubly (or more) embedded complex predicate, is not a weakness on the semantic side but relates to the syntax, and its solution is not specific to the analysis of complex predication, since the problem already affects the analysis of other, considerably more basic, phenomena (like recursive modification). This weakness aside, the present proposal also has the potential to go beyond both the linking and XLE approaches to complex predication in its ability to deal with optionality of arguments. As the only proposal for a semantically integrated account of complex predicates within the current standard approach to LFG + glue and the current standard LFG architecture, its potential for dealing with a wider range of complex predicate phenomena, including phenomena that are problematic for earlier approaches, is a worthy subject for future research.

$$
\dfrac{
\begin{array}{l}
\lambda P.\lambda y.\lambda x.\lambda e.\mathit{let}(x,y,P(y,e)) :\\
[(\uparrow_\sigma \mathrm{ARG}_1) \multimap (\uparrow_\sigma \mathrm{EV}) \multimap \uparrow_\sigma] \multimap\\
(\uparrow_\sigma \mathrm{ARG}_3) \multimap (\uparrow_\sigma \mathrm{ARG}_1) \multimap (\uparrow_\sigma \mathrm{EV}) \multimap \uparrow_\sigma
\end{array}
\quad
\dfrac{
\dfrac{
\dfrac{
\begin{array}{l}
\lambda y.\lambda x.\lambda e.\mathit{pinch}(e) \wedge \mathit{agent}(e,x)\\
\wedge \mathit{patient}(e,y) : (\uparrow_\sigma \mathrm{ARG}_2) \multimap (\uparrow_\sigma \mathrm{ARG}_1)\\
\multimap (\uparrow_\sigma \mathrm{EV}) \multimap \uparrow_\sigma
\end{array}
\quad
\begin{array}{l}
\lambda P.\lambda x.\lambda e.P(x,e) :\\
[(\uparrow_\sigma \mathrm{ARG}_1) \multimap (\uparrow_\sigma \mathrm{EV}) \multimap \uparrow_\sigma] \multimap\\
(\uparrow_\sigma \mathrm{ARG}_1) \multimap (\uparrow_\sigma \mathrm{EV}) \multimap \uparrow_\sigma
\end{array}
}{
\begin{array}{l}
\lambda y.\lambda x.\lambda e.\mathit{pinch}(e) \wedge \mathit{agent}(e,x)\\
\wedge \mathit{patient}(e,y) : (\uparrow_\sigma \mathrm{ARG}_2) \multimap (\uparrow_\sigma \mathrm{ARG}_1)\\
\multimap (\uparrow_\sigma \mathrm{EV}) \multimap \uparrow_\sigma
\end{array}
}
\quad
\begin{array}{l}
\lambda P.\lambda y.\lambda x.\lambda e.\mathit{cause}(x,y,P(y,e)) :\\
[(\uparrow_\sigma \mathrm{ARG}_1) \multimap (\uparrow_\sigma \mathrm{EV}) \multimap \uparrow_\sigma] \multimap\\
(\uparrow_\sigma \mathrm{ARG}_4) \multimap (\uparrow_\sigma \mathrm{ARG}_1) \multimap (\uparrow_\sigma \mathrm{EV}) \multimap \uparrow_\sigma
\end{array}
}{
\begin{array}{l}
\lambda z.\lambda y.\lambda x.\lambda e.\mathit{cause}(x,y,(\mathit{pinch}(e)\\
\wedge \mathit{agent}(e,y) \wedge \mathit{patient}(e,z))) : (\uparrow_\sigma \mathrm{ARG}_2) \multimap (\uparrow_\sigma \mathrm{ARG}_4)\\
\multimap (\uparrow_\sigma \mathrm{ARG}_1) \multimap (\uparrow_\sigma \mathrm{EV}) \multimap \uparrow_\sigma
\end{array}
}
\quad
\begin{array}{l}
\lambda P.\lambda x.\lambda e.\mathit{completely}(P(x,e)) :\\
[(\uparrow_\sigma \mathrm{ARG}_1) \multimap (\uparrow_\sigma \mathrm{EV}) \multimap \uparrow_\sigma] \multimap\\
(\uparrow_\sigma \mathrm{ARG}_1) \multimap (\uparrow_\sigma \mathrm{EV}) \multimap \uparrow_\sigma
\end{array}
}{
\begin{array}{l}
\lambda z.\lambda y.\lambda x.\lambda e.\mathit{completely}(\mathit{cause}(x,y,\\
(\mathit{pinch}(e) \wedge \mathit{agent}(e,y) \wedge \mathit{patient}(e,z)))) :\\
(\uparrow_\sigma \mathrm{ARG}_2) \multimap (\uparrow_\sigma \mathrm{ARG}_4) \multimap (\uparrow_\sigma \mathrm{ARG}_1) \multimap (\uparrow_\sigma \mathrm{EV}) \multimap \uparrow_\sigma
\end{array}
}
}{
\begin{array}{c}
\lambda z.\lambda y.\lambda x.\lambda w.\lambda e.\mathit{let}(w,x,\mathit{completely}(\mathit{cause}(x,y,(\mathit{pinch}(e) \wedge \mathit{agent}(e,y) \wedge \mathit{patient}(e,z))))) :\\
(\uparrow_\sigma \mathrm{ARG}_2) \multimap (\uparrow_\sigma \mathrm{ARG}_4) \multimap (\uparrow_\sigma \mathrm{ARG}_3) \multimap (\uparrow_\sigma \mathrm{ARG}_1) \multimap (\uparrow_\sigma \mathrm{EV}) \multimap \uparrow_\sigma
\end{array}
}
$$

Figure 1: Glue proof for (38)

$$
\dfrac{
  \dfrac{
    \mathit{Amu} : \uparrow_\sigma
    \quad
    \dfrac{
      \mathit{child} : \uparrow_\sigma
      \quad
      \dfrac{
        \dfrac{
          \dfrac{
            \begin{array}{l}
            \lambda y.\lambda x.\lambda e.\mathit{pinch}(e) \wedge \mathit{agent}(e,x) \\
            \wedge \mathit{patient}(e,y) : (\uparrow_\sigma \text{ARG}_2) \multimap (\uparrow_\sigma \text{ARG}_1) \\
            \multimap (\uparrow_\sigma \text{EV}) \multimap \uparrow_\sigma
            \end{array}
            \quad
            \begin{array}{l}
            \lambda P.\lambda x.\lambda e.P(x,e) : \\
            [(\uparrow_\sigma \text{ARG}_1) \multimap (\uparrow_\sigma \text{EV}) \multimap \uparrow_\sigma] \multimap \\
            (\uparrow_\sigma \text{ARG}_1) \multimap (\uparrow_\sigma \text{EV}) \multimap \uparrow_\sigma
            \end{array}
          }{
            \begin{array}{l}
            \lambda y.\lambda x.\lambda e.\mathit{pinch}(e) \wedge \mathit{agent}(e,x) \\
            \wedge \mathit{patient}(e,y) : (\uparrow_\sigma \text{ARG}_2) \multimap (\uparrow_\sigma \text{ARG}_1) \\
            \multimap (\uparrow_\sigma \text{EV}) \multimap \uparrow_\sigma
            \end{array}
          }
          \quad
          \begin{array}{l}
          \lambda P.\lambda y.\lambda x.\lambda e.\mathit{cause}(x,y,P(y,e)) : \\
          [(\uparrow_\sigma \text{ARG}_1) \multimap (\uparrow_\sigma \text{EV}) \multimap \uparrow_\sigma] \multimap \\
          (\uparrow_\sigma \text{ARG}_4) \multimap (\uparrow_\sigma \text{ARG}_1) \multimap (\uparrow_\sigma \text{EV}) \multimap \uparrow_\sigma
          \end{array}
        }{
          \begin{array}{l}
          \lambda z.\lambda y.\lambda x.\lambda e.\mathit{cause}(x,y,(\mathit{pinch}(e) \\
          \wedge \mathit{agent}(e,y) \wedge \mathit{patient}(e,z))) : (\uparrow_\sigma \text{ARG}_2) \multimap (\uparrow_\sigma \text{ARG}_4) \\
          \multimap (\uparrow_\sigma \text{ARG}_1) \multimap (\uparrow_\sigma \text{EV}) \multimap \uparrow_\sigma
          \end{array}
        }
        \quad
        \mathit{elephant} : \uparrow_\sigma
      }{
        \begin{array}{l}
        \lambda y.\lambda x.\lambda e.\mathit{cause}(x,y,(\mathit{pinch}(e) \\
        \wedge \mathit{agent}(e,y) \wedge \mathit{patient}(e,\mathit{elephant}))) : \\
        (\uparrow_\sigma \text{ARG}_4) \multimap (\uparrow_\sigma \text{ARG}_1) \multimap (\uparrow_\sigma \text{EV}) \multimap \uparrow_\sigma
        \end{array}
      }
    }{
      \begin{array}{l}
      \lambda x.\lambda e.\mathit{cause}(x,\mathit{child},(\mathit{pinch}(e) \\
      \wedge \mathit{agent}(e,\mathit{child}) \wedge \mathit{patient}(e,\mathit{elephant}))) : \\
      (\uparrow_\sigma \text{ARG}_1) \multimap (\uparrow_\sigma \text{EV}) \multimap \uparrow_\sigma
      \end{array}
    }
  }{
    \begin{array}{l}
    \lambda e.\mathit{cause}(\mathit{Amu},\mathit{child},(\mathit{pinch}(e) \\
    \wedge \mathit{agent}(e,\mathit{child}) \wedge \mathit{patient}(e,\mathit{elephant}))) : \\
    (\uparrow_\sigma \text{EV}) \multimap \uparrow_\sigma
    \end{array}
  }
  \quad
  \begin{array}{l}
  \lambda P.\exists e.P(e) : \\
  ((\uparrow_\sigma \text{EV}) \multimap \uparrow_\sigma) \multimap \uparrow_\sigma
  \end{array}
}{
  \begin{array}{l}
  \exists e.\mathit{cause}(\mathit{Amu},\mathit{child},(\mathit{pinch}(e) \\
  \wedge \mathit{agent}(e,\mathit{child}) \wedge \mathit{patient}(e,\mathit{elephant}))) : \\
  \uparrow_\sigma
  \end{array}
}
$$

Figure 2: Glue proof for (47)

$$
\dfrac{
\mathit{Amu}:\uparrow_\sigma
\qquad
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\begin{array}{l}
\lambda y.\lambda x.\lambda e.\mathit{pinch}(e)\wedge \mathit{agent}(e,x)\\
\wedge \mathit{patient}(e,y):(\uparrow_\sigma \textsc{arg}_2)\multimap(\uparrow_\sigma \textsc{arg}_1)\\
\multimap(\uparrow_\sigma \textsc{ev})\multimap\uparrow_\sigma
\end{array}
\qquad
\begin{array}{l}
\lambda P.\lambda x.\lambda e.P(x,e):\\
[(\uparrow_\sigma \textsc{arg}_1)\multimap(\uparrow_\sigma \textsc{ev})\multimap\uparrow_\sigma]\multimap\\
(\uparrow_\sigma \textsc{arg}_1)\multimap(\uparrow_\sigma \textsc{ev})\multimap\uparrow_\sigma
\end{array}
}{
\begin{array}{l}
\lambda y.\lambda x.\lambda e.\mathit{pinch}(e)\wedge \mathit{agent}(e,x)\\
\wedge \mathit{patient}(e,y):(\uparrow_\sigma \textsc{arg}_2)\multimap(\uparrow_\sigma \textsc{arg}_1)\\
\multimap(\uparrow_\sigma \textsc{ev})\multimap\uparrow_\sigma
\end{array}
}
\qquad
\begin{array}{l}
\lambda P.\lambda y.\lambda x.\lambda e.\mathit{cause}(x,y,P(y,e)):\\
[(\uparrow_\sigma \textsc{arg}_1)\multimap(\uparrow_\sigma \textsc{ev})\multimap\uparrow_\sigma]\multimap\\
(\uparrow_\sigma \textsc{arg}_4)\multimap(\uparrow_\sigma \textsc{arg}_1)\multimap(\uparrow_\sigma \textsc{ev})\multimap\uparrow_\sigma
\end{array}
}{
\begin{array}{l}
\lambda z.\lambda y.\lambda x.\lambda e.\mathit{cause}(x,y,(\mathit{pinch}(e)\\
\wedge \mathit{agent}(e,y)\wedge \mathit{patient}(e,z))):(\uparrow_\sigma \textsc{arg}_2)\multimap(\uparrow_\sigma \textsc{arg}_4)\\
\multimap(\uparrow_\sigma \textsc{arg}_1)\multimap(\uparrow_\sigma \textsc{ev})\multimap\uparrow_\sigma
\end{array}
}
\qquad
\mathit{elephant}:\uparrow_\sigma
}{
\begin{array}{l}
\lambda y.\lambda x.\lambda e.\mathit{cause}(x,y,(\mathit{pinch}(e)\\
\wedge \mathit{agent}(e,y)\wedge \mathit{patient}(e,\mathit{elephant}))):\\
(\uparrow_\sigma \textsc{arg}_4)\multimap(\uparrow_\sigma \textsc{arg}_1)\multimap(\uparrow_\sigma \textsc{ev})\multimap\uparrow_\sigma
\end{array}
}
\qquad
\begin{array}{l}
\lambda P.\exists x.P(x):\\
((\uparrow_\sigma \textsc{arg}_4)\multimap\uparrow_\sigma)\multimap\uparrow_\sigma
\end{array}
}{
\begin{array}{l}
\lambda x.\lambda e.\exists y.\mathit{cause}(x,y,(\mathit{pinch}(e)\\
\wedge \mathit{agent}(e,y)\wedge \mathit{patient}(e,\mathit{elephant}))):\\
(\uparrow_\sigma \textsc{arg}_1)\multimap(\uparrow_\sigma \textsc{ev})\multimap\uparrow_\sigma
\end{array}
}
}{
\dfrac{
\begin{array}{l}
\lambda e.\exists y.\mathit{cause}(\mathit{Amu},y,(\mathit{pinch}(e)\\
\wedge \mathit{agent}(e,y)\wedge \mathit{patient}(e,\mathit{elephant}))):\\
(\uparrow_\sigma \textsc{ev})\multimap\uparrow_\sigma
\end{array}
\qquad
\begin{array}{l}
\lambda P.\exists e.P(e):\\
((\uparrow_\sigma \textsc{ev})\multimap\uparrow_\sigma)\multimap\uparrow_\sigma
\end{array}
}{
\begin{array}{l}
\exists e.\exists y.\mathit{cause}(\mathit{Amu},y,(\mathit{pinch}(e)\\
\wedge \mathit{agent}(e,y)\wedge \mathit{patient}(e,\mathit{elephant}))):\\
\uparrow_\sigma
\end{array}
}
}
$$

Figure 3: Glue proof for (48)

REFERENCES

Farrell ACKERMAN and Gert WEBELHUTH (1996), The Construct PREDICATE: Empirical Arguments and Theoretical Status, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG96 Conference*, CSLI Publications, Stanford, CA, http://cslipublications.stanford.edu/LFG/1/ackerman.html.

Farrell ACKERMAN and Gert WEBELHUTH (1998), *A Theory of Predicates*, CSLI Publications, Stanford, CA.

Tafseer AHMED and Miriam BUTT (2011), Discovering Semantic Classes for Urdu N-V Complex Predicates, in *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*, pp. 305–309, Association for Computational Linguistics, Stroudsburg, PA.

Tafseer AHMED, Miriam BUTT, Annette HAUTLI, and Sebastian SULGER (2012), A Reference Dependency Bank for Analyzing Complex Predicates, in Nicoletta CALZOLARI, Khalid CHOUKRI, Thierry DECLERCK, Mehmet UĞUR DOĞAN, Bente MAEGAARD, Joseph MARIANI, Jan ODIJK, and Stelios PIPERIDIS, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, European Language Resources Association (ELRA).

Alex ALSINA (1996), *The Role of Argument Structure in Grammar*, CSLI Publications, Stanford, CA.

Alex ALSINA (1997), A theory of Complex Predicates: Evidence from Causatives in Bantu and Romance, in Alex ALSINA, Joan BRESNAN, and Peter SELLS, editors, *Complex Predicates*, pp. 203–246, CSLI Publications, Stanford, CA.

Alex ALSINA and Smita JOSHI (1991), Parameters in Causative Constructions, in L. DOBRIN, L. NICHOLS, and R. M. RODRIGUEZ, editors, *Proceedings from the 27th Regional Meeting of the Chicago Linguistic Society (CLS)*, pp. 1–16, Chicago Linguistic Society, Chicago, IL.

Avery D. ANDREWS (2001), Iofu and Spreading Architecture in LFG, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG01 Conference*, CSLI Publications, Stanford, CA.

Avery D. ANDREWS (2007), Projections and Glue for Clause-Union Complex Predicates, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG07 Conference*, pp. 44–65, CSLI Publications, Stanford, CA.

Avery D. ANDREWS (2008), The Role of PRED in LFG + Glue, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG08 Conference*, pp. 47–67, CSLI Publications, Stanford, CA.

Avery D. ANDREWS (2010), Propositional Glue and the Projection Architecture of LFG, *Linguistics and Philosophy*, 33(3):141–170.

Avery D. ANDREWS (2015), Sets and Heads in LFG, MS, Australian National University.

Avery D. ANDREWS and Christopher D. MANNING (1999), *Complex Predicates and Information Spreading in LFG*, CSLI Publications, Stanford, CA.

Ash ASUDEH (2004), *Resumption as Resource Management*, Ph.D. thesis, Stanford University.

Ash ASUDEH (2009), Adjacency and Locality: a Constraint-Based Analysis of Complementizer-Adjacent Extraction, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG09 Conference*, pp. 106–126, CSLI Publications, Stanford, CA.

Ash ASUDEH (2012), *The Logic of Pronominal Resumption*, Oxford University Press, Oxford.

Ash ASUDEH and Richard CROUCH (2002), Coordination and Parallelism in Glue Semantics: Integrating Discourse Cohesion and the Element Constraint, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG02 Conference*, CSLI Publications, Stanford, CA.

Ash ASUDEH, Mary DALRYMPLE, and Ida TOIVONEN (2008), Constructions with Lexical Integrity: Templates as the Lexicon-Syntax Interface, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG08 Conference*, pp. 68–88, CSLI Publications, Stanford, CA.

Ash ASUDEH, Mary DALRYMPLE, and Ida TOIVONEN (2013), Constructions with Lexical Integrity, *Journal of Language Modelling*, 1(1):1–54.

Ash ASUDEH and Gianluca GIORGOLO (2012), Flexible Composition for Optional and Derived Arguments, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG12 Conference*, pp. 64–84, CSLI Publications, Stanford, CA.

Ash ASUDEH, Gianluca GIORGOLO, and Ida TOIVONEN (2014), Meaning and Valency, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG14 Conference*, pp. 68–88, CSLI Publications, Stanford, CA.

Joan BRESNAN (2001), *Lexical-Functional Syntax*, Blackwell, Oxford.

Joan BRESNAN and Jonni M. KANERVA (1989), Locative Inversion in Chicheŵa: A Case Study of Factorization in Grammar, *Linguistic Inquiry*, 20:1–50.

Miriam BUTT (1994), Machine Translation and Complex Predicates, in Harald TROST, editor, *Proceedings of KONVENS '94 "Verarbeitung natürlicher Sprache", Vienna, September 28–30, 1994*, pp. 62–71, Springer, Berlin.

Miriam BUTT (1995), *The Structure of Complex Predicates in Urdu*, CSLI Publications, Stanford, CA.

Miriam BUTT (1997), Complex Predicates in Urdu, in Alex ALSINA, Joan BRESNAN, and Peter SELLS, editors, *Complex Predicates*, CSLI Publications, Stanford, CA.

Miriam BUTT (1998), Constraining Argument Merger Through Aspect, in Erhard HINRICHS, Andreas KATHOL, and Tsuneko NAKAZAWA, editors, *Complex Predicates in Nonderivational Syntax*, pp. 73–113, Academic Press, San Diego, CA.

Miriam BUTT (2014), Control vs. Complex Predication: Identifying Non-Finite Complements, *Natural Language & Linguistic Theory*, 32(1):165–190.

Miriam BUTT, Tina BÖGEL, Annette HAUTLI, Sebastian SULGER, and Tafseer AHMED (2012), Identifying Urdu Complex Predication via Bigram Extraction, in *Proceedings of the 24th International Conference on Computational Linguistics (COLING12)*, pp. 409–424.

Miriam BUTT, Helge DYVIK, Tracy Holloway KING, Hiroshi MASUICHI, and Christian ROHRER (2002), The Parallel Grammar Project, in J. CARROLL, N. OOSTIJK, and R. SUTCLIFFE, editors, *Proceedings of COLING 2002: Workshop on Grammar Engineering and Evaluation*, pp. 1–7.

Miriam BUTT and Wilhelm GEUDER (2001), On the (Semi)lexical Status of Light Verbs, in Norbert CORVER and Henk VAN RIEMSDIJK, editors, *Semi-Lexical Categories: On the Content of Function Words and the Function of Content Words*, pp. 323–370, Mouton de Gruyter, Berlin.

Miriam BUTT and Tracy Holloway KING (2006), Restriction for Morphological Valency Alternations: The Urdu Causative, in Miriam BUTT, Mary DALRYMPLE, and Tracy Holloway KING, editors, *Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan*, pp. 235–358, CSLI Publications, Stanford, CA.

Miriam BUTT and Tracy Holloway KING (2007), Urdu in a Parallel Grammar Development Environment, *Language Resources and Evaluation (Special Issue on Asian Language Processing: State of the Art Resources and Processing)*, 41:191–207.

Miriam BUTT, Tracy Holloway KING, and John T. MAXWELL, III (2003), Complex Predicates via Restriction, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG03 Conference*, CSLI Publications, Stanford, CA.

Miriam BUTT, Tracy Holloway KING, María-Eugenia NIÑO, and Frédérique SEGOND (1999), *A Grammar Writer's Cookbook*, CSLI Publications, Stanford, CA.

Miriam BUTT, Tracy Holloway KING, and Gillian RAMCHAND (2010), Complex Predication: How Did the Child Pinch the Elephant?, in Linda UYECHI and Lian Hee WEE, editors, *Reality Exploration and Discovery: Pattern Interaction in Language & Life. A Festschrift for K. P. Mohanan*, pp. 231–256, CSLI Publications, Stanford, CA.

Miriam BUTT, María-Eugenia NIÑO, and Frédérique SEGOND (1996), Multilingual Processing of Auxiliaries within LFG, in Dafydd GIBBON, editor, *Natural Language Processing and Speech Technology: Results of the 3$^{\text{rd}}$ KONVENS Conference, Bielefeld, October 1996*, pp. 111–122, Mouton de Gruyter, Berlin.

Miriam BUTT and Gillian RAMCHAND (2005), Complex Aspectual Structure in Hindi/Urdu, in Nomi ERTESCHIK-SHIR and Tova RAPOPORT, editors, *The Syntax of Aspect*, pp. 117–153, Oxford University Press, Oxford.

Liz CHRISTIE (2013), Result XPs and the Argument-Adjunct Distinction, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG13 Conference*, pp. 212–231, CSLI Publications, Stanford, CA.

Dick CROUCH, Mary DALRYMPLE, Ronald M. KAPLAN, Tracy Holloway KING, John T. MAXWELL III, and Paula NEWMAN (2011), *XLE Documentation*, Palo Alto Research Center, Palo Alto, CA, `http://www2.parc.com/isl/groups/nltt/xle/doc/xle_toc.html`.

Dick CROUCH and Tracy Holloway KING (2006), Semantics via F-structure Rewriting, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG06 Conference*, CSLI Publications, Stanford, CA.

Mary DALRYMPLE (2001), *Lexical Functional Grammar*, Academic Press, San Diego, CA.

Mary DALRYMPLE, Vineet GUPTA, John LAMPING, and Vijay SARASWAT (1999), Relating Resource-based Semantics to Categorial Semantics, in Mary DALRYMPLE, editor, *Semantics and Syntax in Lexical Functional Grammar*, pp. 261–280, MIT Press, Cambridge, Mass., also in *Proceedings of the Fifth Meeting on Mathematics of Language* (MOL5), Schloss Dagstuhl, Saarbrücken, Germany, August 1997.

Mary DALRYMPLE, Angie HINRICHS, John LAMPING, and Vijay SARASWAT (1993a), The Resource Logic of Complex Predicate Interpretation, in K.-J. CHEN and C.-R. HUANG, editors, *Proceedings of the 1993 Republic of China Computational Linguistics Conference (ROCLING)*, Computational Linguistics Society of the Republic of China, Hsitou National Park, Taiwan, also published as Xerox Technical Report ISTL-NLTT-1993-08-03.

Mary DALRYMPLE, Ronald M. KAPLAN, and Tracy Holloway KING (2004), Linguistic Generalizations over Descriptions, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG04 Conference*, pp. 199–208, CSLI Publications, Stanford, CA.

Mary DALRYMPLE, John LAMPING, Fernando PEREIRA, and Vijay SARASWAT (1996), A Deductive Account of Quantification in LFG, in Makoto KANAZAWA, Christopher J. PIÑÓN, and Henriette DE SWART, editors, *Quantifiers, Deduction and Context*, pp. 33–57, CSLI Publications, Stanford, CA.

Mary DALRYMPLE, John LAMPING, and Vijay SARASWAT (1993b), LFG Semantics via Constraints, in *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*, pp. 97–105, Association for Computational Linguistics.

Yehuda N. FALK (2001), *Lexical-Functional Grammar: an Introduction to Parallel Constraint-Based Syntax*, CSLI Publications, Stanford, CA.

Jamie FINDLAY (2014), Mapping Theory without Argument Structure, MS, University of Oxford.

John FRY (2005), Resource-logical Event Semantics for LFG, draft MS, based on a paper originally presented at LFG99, University of Manchester. `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.6503&rep=rep1&type=pdf`.

Dag T. T. HAUG (2008), Tense and Aspect for Glue Semantics: The Case of Participial XADJ's, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG08 Conference*, pp. 291–311, CSLI Publications, Stanford, CA.

Petr HOMOLA and Matt COLER (2013), Causatives as Complex Predicates without the Restriction Operator, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG13 Conference*, pp. 316–334, CSLI Publications, Stanford, CA.

Akira ISHIKAWA (1985), *Complex Predicates and Lexical Operations in Japanese*, Ph.D. thesis, Stanford University.

Ronald M. KAPLAN and Joan BRESNAN (1982), Lexical-Functional Grammar: A Formal System for Grammatical Representation, in Joan BRESNAN, editor, *The Mental Representation of Grammatical Relations*, pp. 173–281, MIT Press, Cambridge, MA.

Ronald M. KAPLAN and Jürgen WEDEKIND (1993), Restriction and Correspondence-Based Translation, in *Proceedings of the 6th Meeting of the EACL*, pp. 193–202, European Association for Computational Linguistics, Utrecht.

Anna KIBORT (2001), The Polish Passive and Impersonal in Lexical Mapping Theory, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG01 Conference*, pp. 163–183, CSLI Publications, Stanford, CA.

Anna KIBORT (2004), *Passive and Passive-Like Constructions in English and Polish*, Ph.D. thesis, University of Cambridge.

Anna KIBORT (2006), On Three Different Types of Subjectlessness and How to Model Them in LFG, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG06 Conference*, CSLI Publications, Stanford, CA.

Anna KIBORT (2007), Extending the Applicability of Lexical Mapping Theory, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG07 Conference*, pp. 250–270, CSLI Publications, Stanford, CA.

Anna KIBORT (2008), On the Syntax of Ditransitive Constructions, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG08 Conference*, pp. 312–332, CSLI Publications, Stanford, CA.

Miltiadis KOKKONIDIS (2008), First-Order Glue, *Journal of Logic, Language and Information*, 17:43–68.

Jonas KUHN (2001), Resource Sensitivity in the Syntax-Semantics Interface and the German Split NP Construction, in W. Detmar MEURERS and Tibor KISS, editors, *Constraint-Based Approaches to Germanic Syntax*, pp. 177–216, CSLI Publications, Stanford, CA.

John J. LOWE (2014), Gluing Meanings and Semantic Structures, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG14 Conference*, pp. 387–407, CSLI Publications, Stanford, CA.

John J. LOWE (2015), *Participles in Rigvedic Sanskrit: The Syntax and Semantics of Adjectival Verb Forms*, Oxford University Press, Oxford.

Tara MOHANAN (1994), *Argument Structure in Hindi*, CSLI Publications, Stanford, CA.

Stephanie NEEDHAM and Ida TOIVONEN (2011), Derived Arguments, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG11 Conference*, pp. 401–421, CSLI Publications, Stanford, CA.

Ghulam RAZA (2011), *Subcategorization Acquisition and Classes of Predication in Urdu*, Ph.D. thesis, University of Konstanz.

Sebastian SULGER (2012), Nominal Argument Structure and the Stage-/Individual-Level Contrast in Hindi-Urdu, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG12 Conference*, pp. 582–602, CSLI Publications, Stanford, CA.

Sebastian SULGER, Miriam BUTT, Tracy Holloway KING, Paul MEURER, Tibor LACZKÓ, György RÁKOSI, Cheikh Bamba DIONE, Helge DYVIK, Victoria ROSÉN, Koenraad DE SMEDT, Agnieszka PATEJUK, Özlem ÇETINOĞLU, I Wayan ARKA, and Meladel MISTICA (2013), ParGramBank: The ParGram Parallel Treebank, in *Proceedings of 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, August 4–9, 2013*, pp. 550–560, Association for Computational Linguistics.

Ida TOIVONEN (2013), English Benefactive NPs, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG13 Conference*, pp. 503–523, CSLI Publications, Stanford, CA.

Annie ZAENEN and Mary DALRYMPLE (1995), “Polymorphic” Causatives: Complex Predicates in French, Technical report, Information Sciences and Technologies Laboratory, Xerox PARC, Palo Alto, CA.

Annie ZAENEN and Mary DALRYMPLE (1996), Les Verbes Causatifs “Polymorphiques”: Les Prédicats Complexes en Français, *Langages*, 122:79–95.

Representing morphological tone in a computational grammar of Hausa

Berthold Crysmann
CNRS, Laboratoire de linguistique formelle (UMR 7110), Paris

ABSTRACT

In this paper[1] I shall discuss the representation of morphological tone in Hausa, as implemented in a computational grammar of the language, referred to as HAG, which has been developed within the framework of Head-driven Phrase Structure Grammar. Based on an in-depth study of segmental and suprasegmental properties manipulated by morphological processes, I shall argue that two fundamental insights from autosegmental phonology need to be seamlessly integrated into typed feature structure grammars of languages with grammatical tone, namely (i) the systematic separation of tonal and metrical information from the string of consonants and vowels, and (ii) the possibility of tonal spreading, i.e. the possibility for a tonal specification to be

Keywords: Hausa, HPSG, Autosegmental Phonology, tone, implementation

[1] A previous version of this work was presented at the Workshop on "Disentangling the inflectional role of tone", organised by Jean-Léo Léonard (U Paris Sorbonne & Institut Universitaire de France) and Enrique Palancár (CNRS, SeDyl–CELIA) at the Maison de Recherche, Paris 3 (Sorbonne Nouvelle) in June 2013, as well as at a research seminar on tone and intonation at the Institut für Linguistik, Phonetik, Universität zu Köln. I would therefore like to thank the respective audiences for their helpful and stimulating comments and, in particular, Larry Hyman, Jean-Léo Léonard, Enrique Palancár, Martine Grice, Phuong Ha, and Doris Mücke. A great many thanks also go to the three anonymous reviewers whose comments and suggestions have been invaluable in improving this paper. Finally, I would also like to express my gratitude to Carmela Chateau for her careful proofing and copy-editing.

Work on the development of the grammar is partially supported by the French National Research Agency (ANR) as part of the "Investissements d'Avenir" programme (reference: ANR-10-LABX-0083), project ResHau [re:'zo:] of the Laboratoire d'excéllence "Empirical Foundations of Linguistics".

Journal of Language Modelling Vol 3, No 2 (2015), pp. 463–512

assigned to an arbitrary number of adjacent tone-bearing units (syllables). To this end, I present a formalisation of tonal melodies in terms of typed list constraints that implement a notion of tonal spreading, allowing for an underspecified description of tonal melodies, independent of the number of tone-bearing units. I shall finally show that this minimal encoding is sufficient, and flexible enough to capture the range of suprasegmental phenomena in Hausa.

1 INTRODUCTION

One of the fundamental motivations for the implementation of grammars in linguistically motivated formalisms is to allow for rigorous testing of the empirical predictions of linguistic theories: given the complex interactions of rich lexica with highly general principles and rules, paired with increasing theoretical coverage of phenomena way past the core into the periphery, manual evaluation of the consequences of a theory is becoming less and less feasible. By running an implemented grammar over a corpus, it is possible not only to detect limitations in coverage, but also to assess the degree of overgeneration. This latter aspect is particularly easy to ensure in the context of reversible grammars, i.e. declarative knowledge sources that can be used for both parsing and generation. Despite the widespread use of tone to mark lexical and grammatical distinctions among the languages of the world, none of the grammars implemented to date actually incorporates a treatment of suprasegmentals into the grammar proper. This is due to the fact that, in the sample of languages for which implemented grammars exist, very few are actually tone languages, and the ones that are, like Chinese, only have lexical, not grammatical, tone.

Within theoretical linguistics, by contrast, the study of tone has always enjoyed a more central role, at least over the past 45 years. Investigation into African tone languages (Goldsmith 1976; Leben 1973) has provided the main evidence for the development of multi-tiered approaches to phonology, culminating in the development of autosegmental metrical phonology, subsequently to be adopted for the treatment of supra-laryngeal phenomena as well (Clements 1985). Due to the transformational heritage (Chomsky and Halle 1968), however, some of the descriptive devices that are used to extract generalisations about tonal phenomena, especially destructive operations or procedural notions such as pre-linking, de-linking and re-spreading, do not lend themselves naturally to direct integration into the kind of formalisms used in the development of linguistically motivated computational grammars, which are generally committed to monotonicity and declarativity.[2] As established by work on Declarative or One-Level Phonology (Bird and Klein 1994; Scobbie 1993), typed feature structures, as used in HPSG (Pollard and Sag 1987, 1994), provide a convenient and expressive representation for capturing multi-tiered phonological descriptions. However, no sizeable grammar fragment of a tone language has so far been developed within a declarative phonological framework, let alone in the context of implemented computational grammars.

In this paper, I seek to close this gap between grammar engineering and theoretical autosegmental description. In particular, based on the example of Hausa, a language featuring both lexical and grammatical tone, I shall argue that the adoption of an autosegmental approach, i.e. one that separates the representation of tone and vowel length from that of consonantal and vocalic segments, is not only preferable for theoretical reasons, but actually inescapable from a grammar-engineering perspective. Furthermore, I shall show how the intimate connection between suprasegmental phenomena and morphosyntax can be accommodated in TDL (=Type Description Language; Copestake 2002; Krieger 1996), the purely conjunctive and monotonic typed feature formalism underlying HAG. Besides serving the practical purpose of tight integration with morphology and syntax in the implemented grammar, the choice of a lean formalism enables us to explore whether the general formalism is expressive enough to capture the relevant generalisations. More specifically, I shall argue that when suprasegmental operations are aligned with morphological rules, the available mechanisms, though highly restricted by themselves, will nevertheless prove to be sufficient to capture the entire set of surface-true and surface-apparent generalisations on Hausa suprasegmental

[2] Frameworks differ, of course, as to the extent of uniformity they assume across different linguistic sub-theories: while HPSG defends the hypothesis that all levels of linguistic knowledge (phonological, morphological, syntactic, and semantic) should be expressible in the same formalism, projection-architectures like LFG do not subscribe to this assumption.

phonology. The restrictive nature of our formalisation, which only ever permits one spreading tone per morphological domain, will prove to have interesting theoretical consequences: first, I resolve the debate regarding the direction of tone assignment between right-to-left (Newman 2000) and left-to-right (Leben 1978) in favour of outside-in, with a predominance, in the case of Hausa, of assignment from the right edge. Second, I shall show that all cases with more than one spreading tone in Hausa involve total reduplication, and argue that the proper treatment of this phenomenon independently requires morphological compounding, giving rise to two independent domains, each with a spreading tone of its own.

The paper is organised as follows: in Section 2, I shall give a general overview of Hausa tone, followed by a detailed study of morphological tone in Section 3, capitalising on holistic assignment of tone melodies vs. agglutinative tone. Section 4 provides an overview of HaG, an implemented HPSG of Hausa that crucially integrates suprasegmental information with morphological and syntacto-semantic analysis. In Section 5, I shall present the autosegmental representation as implemented in HaG, capitalising on tonal spreading, as well as morphological operations on tone and length, including holistic assignment of tonal melodies and local adjustments of tone and length specifications. A major part of the discussion in this section will be concerned with the integration of prespecified prefixal tone, and its integration with right-to-left spreading, using conjunctive tone list constraints (Section 5.4.2), while Section 5.5 closes with a general discussion regarding the expressive power of the current approach.

2 SUPRASEGMENTAL DISTINCTIONS IN HAUSA

On the suprasegmental level, both tone and vowel length are distinctive. Hausa recognises two level tones, H(igh) and L(ow), as well as a falling contour tone, which is typically analysed as an HL sequence associated with a single (heavy) syllable. Rising contours observable at the surface are the result of interaction between lexical or grammatical tone and intonation (Inkelas and Leben 1990).[3] Throughout

[3] LH sequences associated with a single syllable, e.g. as a result of affixation, undergo obligatory tone simplification (Newman 2000).

this paper, high tone (H) is marked with an acute accent, low tone (L) with a grave, and falling tone (HL) with a circumflex. On the metrical side, Hausa distinguishes between long and short vowels. Long vowels are marked with a macron, whereas vowels unmarked for length are short. Redundant marking of both H and L was chosen to improve readability for scholars from different traditions: in fact, redundant marking of H deviates from common Hausaist practice (see Section 4 for an overview of the range of conventions found in the literature).[4]

Tone in Hausa serves to distinguish both lexical and grammatical meaning: as shown in (1), the lexical meanings associated with the segmental sequence /fari:/ are distinguished by L-H, H-L, and H-H melodies.

(1) a. *fàrī́* (L H) – ‘look (n)’
b. *fárī̀* (H L) – ‘dry season’
c. *fárī̄́* (H H) – ‘white/whiteness’ (Wolff 1993, p. 56)

Similarly, grammatical distinctions, such as different TAM (= Tense, Aspect, Mood) categories, are equally distinguished by tonal means, as shown in (2), illustrating H, L, and HL (= fall).

(2) a. *yá zṓ* (H H) – he came (relative completive)
b. *yà zṓ* (L H) – he should come (subjunctive)
c. yā́ zṓ (H H) – he came (absolute completive)
d. yā̂ zṓ (HL H) – he might come (potential)

Alongside tone, vowel length is another distinctive suprasegmental property: again, we find minimal pairs, where length serves to distinguish lexical or grammatical meanings, cf. (3) and (4), respectively.

[4] Segmental material, i.e. sequences of consonants and vowels, is represented in standard *Boko* (= Latin script) orthography: hooked letters (*ɓ, ɗ, ƙ*) represent glottalised stops, bigraph *ts* stands for a glottalised alveolar fricative, whereas *’y* denotes a palatalised glottal stop. Other deviations from IPA conventions are *j* (voiced palatal affricate), *c* (voiceless alveo-palatal affricate), and *y* (voiced palatal fricative). The vowel letters *i* and *u* in coda position are actually glides. Geminates are represented by reduplication of the initial letter of a monograph or bi-graph, e.g. *ssh* (= *ʃʃ*) for geminated *sh* (= *ʃ*) or *tts* for a geminated glottalised alveolar fricative (= s’).

(3) a. *fā́sā̀* (CVVCVV) – ‘postpone’
b. *fásā̀* (CVCVV) – ‘smash’ (Newman 2000, p. 400)

(4) a. yā́ zṓ – he came (absolute completive)
b. yá zṓ – he came (relative completive)

Syllables in Hausa are either light (CV) or heavy (CVC or CVV). Thus, long vowels can only be observed in open syllables. The distribution of tone is also sensitive to syllable structure, ruling out any occurrence of the HL contour tone on light CV syllables.

3 TONE AND INFLECTIONAL MORPHOLOGY IN HAUSA

Owing to its lexical and grammatical functions, Hausa tone is intimately linked to morphological operations, both inflectional and derivational. From a tonal perspective, morphological operations can be classified into two types: *agglutinative* and *holistic*. With agglutinative tone assignment, morphological rules simply add a tone to the base, typically together with some segmental material, whereas with holistic assignment, they may specify a completely new melody for the entire base, thereby overwriting lexical tone specifications. Newman (2000) regards tone assignment as a property of the affixes, and therefore distinguishes between tone-integrating affixes, which holistically affect the tonal make-up of the base, and non-integrating affixes, which leave the tones of the base by-and-large unaffected.[5]

3.1 *Holistic tone assignment: tone-integrating suffixes*

Holistic assignment of tonal patterns by morphological operations can probably best be illustrated by Hausa plural formation: as detailed in Table 1, adapted from Newman (2000, p. 431), the overwhelming majority of Hausa plural formation patterns do indeed feature holistic tone assignment. In fact, among the 15 major plural classes, only

[5] Non-integrating affixes may, of course, locally affect the tones at the juncture, as an instance of internal sandhi. What is crucial to the distinction here, is that agglutinative assignment is an entirely local operation, leaving most of the base’s lexical tone intact, whereas holistic assignment discards the base’s lexical tone specification altogether.

class X and XII preserve the tonal melody of the base. All others assign a melody specific to the plural class to bases of heterogeneous tonal make-up (see the detailed discussion of Class I and II below). As a general convention, I shall represent arbitrarily long sequences of like tones, i.e. the result of tone spreading, using the Kleene plus, which denotes arbitrarily many repetitions of the preceding symbol.

Table 1: Hausa plural formation patterns

Class	Plural pattern		Example		Gloss
	Segmental	Tonal	Singular	Plural	
I	-oCi	H$^+$	tā́gā̀	tā́gṓgī́	window
II	-ai	L$^+$-H	dàlī́lī̀	dàlī̀lái	reason
III	-aCe	H$^+$-L-H	dámṓ	dámā̀mḗ	land monitor
IV	-(a)Ca	H-L-H	sírdī́	sírā̀dā́	saddle
V	-aCu	H-L-H	gúrgū̀	gúrā̀gū́	cripple
VI	-uCa	H$^+$-L	hū̀lā́	hū́lúnā̀	cap
VII	-aCi	L$^+$-H	fùrḗ	fùrànnī́	flower
VIII	-aCCaCi	H-L-H-H	gúntū́	gúntàttákī́	stub
IX	-u/-i	L$^+$-H	kújḕrā́	kùjḕrū́	chair
X	-V	–	kwā̀ɗṓ	kwā̀ɗī́	frog
XI	-āwā	L$^+$-H / H$^+$	bàdū́kū̀	dū̀kā̀wā́	leather worker
XII	RED	–	jōjì	jṓjì-jṓjì	judge
XIII	-e + RED	L$^+$-H L$^+$-H	tsírṑ	tsìré-tsìré	shoot/sprout
XIV	RED	H$^+$ L$^+$	mākēkḕ	mā́kā́-mā̀kā̀	expansive
XV	-ī + RED	H$^+$ H$^+$	mī̀nī́nī̀	mínī́-mínī́	tiny

At the segmental level, Hausa employs several different marking devices: suffixation of vowels, with or without reduplication and gemination of the final root consonant (C), as well as total reduplication (RED; classes XII–XV).

The data in (5) provide examples from the highly regular and productive class I nouns,[6] which form their plural by affixation of *-ōCī*, where C represents reduplication of the last root consonant. Base-final vowels, if any, are replaced by the first vowel of the suffix.

[6] I adopt the classification of Newman (2000). For alternative analyses of the Hausa plural system, see e.g. Jaggar (2001) and Wolff (1993).

(5) -ṓCī́ (H$^+$) (Class I) (Newman 2000, p. 432)

a. gúlā̀ (H-L) – gúlṓlī́ (H-H-H)
'drum stick'

b. tā́gā̀ (H-L) – tā́gṓgī́ (H-H-H)
'window'

c. gyàlè (L-L) – gyálṓlī́ (H-H-H)
'shawl'

d. tàmbáyā̀ (L-H-L) – támbáyṓyī́ (H-H-H-H)
'question'

e. kámfànī́ (H-L-H) – kámfánṓnī́ (H-H-H-H)
'company'

f. kwàmìtī̂ (L-L-HL) – kwámítṓcī́ (H-H-H-H)
'committee'

Together with the segmental change, plurals in this highly productive inflectional class are characterised by an all-high tonal melody assigned across the base and the plural affix *-ōCī*. Note further that this tonal assignment is independent of the lexical tone of the base (we find H-L, L-L, L-H-L, and H-L-H). Furthermore, the assignment of an all-high melody appears to be independent of the number of tone-bearing units, there being no difference between trisyllabic and quadrisyllabic words of this plural class. This independence of tonal assignment from the segmental make-up of the word favours an intensional description over an extensional enumeration of tone patterns. One possible description would be assignment of an H tone spreading across the entire domain, as assumed in autosegmental phonology (Leben 1973; Goldsmith 1976).

Another, slightly more complex case of holistic assignment is contributed by class II plurals:

(6) -ái (L$^+$H) (Class II) (Newman 2000, pp. 434–435)

a. àlhájī̀ (L-H-L) – àlhā̀zái (L-L-H)
'Hadji'

b. ɗā́lìbī́ (H-L-H) – ɗā̀lìbái (L-L-H)
'pupil'

c. sánkácḕ (H-H-L) – sànkàtái (L-L-H)
'reaped corn laid down in a row'

d. àlmùbázzàrī́ (L-L-H-L-H) – àlmùbàzzàrái (L-L-L-L-H)
'spendthrift'

e. ɗámī̀ (H-L) – ɗàmmái (L-H)
'bundle'

Here, plural formation adds a suffix *-ai*, and assigns a final LH melody, with the L spreading to the left of the word. As before, assignment of the plural tone pattern is independent of the lexical tone of the base (we find H-L, L-H, H-L-H, L-H-L, and L-H-L-H). Similarly, the very same tonal pattern is assigned to bisyllabic, trisyllabic, and pentasyllabic plurals alike, showing even more clearly the independence of melody specification from the syllable count.

Although the need for the incorporation of spreading is illustrated most clearly in the case of inflectional morphology, where the number of tone-bearing units cannot be established a priori, spreading can also be fruitfully put to use to simplify the inventory of lexical tone melodies. Assuming with Newman (2000) that the first tone of any melody automatically spreads to the left, complex patterns such as H-H-L in (6c), and simpler patterns such as H-L in (6e), can be generalised to H^{+}-L.

3.2 Tonal affixation: non-integrating suffixes

Alongside holistic tone assignment by morphological operations, Hausa also recognises agglutinative tone, i.e., where a tonally specified affix is simply added to a base, leaving the tones of the base fully intact.

An example of a tonally purely agglutinative process is contributed by possessive marking:[7] as shown in (7), Hausa bound possessives are formed by affixation of a consonantal "linker" that agrees in gender and number with the possessum (feminine singular *-r* vs. *-n* otherwise), plus a pronominal affix, marking person, number and gender of the possessor (e.g. third singular feminine *-tà*).

(7) a. ƙwái (H) – ƙwá-n-tà (H-L)
egg(M) egg(M)-L.M-3.S.F
'(her) egg'

[7] See Crysmann (2011) for detailed arguments as to why the Hausa linker and possessive markers should be regarded as morphologically attached affixes, rather than postlexical clitics.

b. rī̀gā́ (L-H) – rī̀gá-r-tà (L-H-L)
gown(F) gown(F)-L.F.SG-3.S.F
‘(her) gown’

c. mṓtā̀ (H-L) – mṓtà-r-tà (H-L-L)
car(F) car(F)-L.F.SG-3.S.F
‘(her) car’

d. kâi (HL) – kâ-n-tà (HL-L)
head(M) head(M)-L.M-3.S.F
‘(her) head’

The possessive pronominal affix *-tà* comes with its own fixed low tone; the tonal specification of the base, however, remains unaffected. Although the linker, which syllabifies with the coda of the base, triggers shortening of the rhyme, in accordance with Hausa’s ban on super-heavy syllables, it leaves the tonal make-up of the rhyme unaffected, as witnessed by the falling tone in (7d).

A slightly more complex case of tonal agglutination can be observed with the specificity or previous reference marker *-r̀/-ǹ*. Segmentally identical to the linker, this consonantal marker adds a low tone to the coda of the base, turning a final H level tone into an HL contour tone, while leaving low-final bases unchanged.

(8) a. ƙwái (H) – ƙwâ-n (HL)
‘the (aforementioned) egg (M)’

b. rī̀gā́ (L-H) – rī̀gâ-r (L-HL)
‘the (aforementioned) gown (F)’

c. mṓtā̀ (H-L) – mṓtà-r (H-L)
‘the (aforementioned) car (F)’

d. kâi (HL) – kâ-n (HL)
‘the (aforementioned) head (M)’

However, in contrast to tone-integrating affixes, the tonal effects on the base are highly local in nature.

The last case of non-integrating affixes I shall consider involves a floating tone. Weak verbal nouns in Hausa are formed by affixation of a long H suffix -*`wā́*. With H-final bases in (9a–c), affixation of -*`wā́* gives rise to an HL contour tone on the preceding heavy syllable (CVV or CVC), a property that can be traced to the marker’s initial low floating tone.

(9) a. káràntā́ (H-L-H) – káràntā̂wā́ (H-L-HL-H)
‘read’ (GR1)

b. sáyár (H-H) – sáyârwā́ (H-HL-H)
‘sell’ (GR5)

c. kā́wṓ (H-H) – kā́wō̂wā́ (H-HL-H)
‘come’ (GR6)

d. kā́mā̀ (H-L) – kā́mā̀wā́ (H-L-H)
‘catch’ (GR1)

e. gyā̀rú (L-H) – gyā̀rúwā́ (L-H-H)
‘be repaired’ (GR7)

However, with light final syllables (as in grade 7;[8] cf. (9e)), the floating low tone is suppressed, reflecting the phonotactic constraint of the language that restricts contour tones to heavy syllables (CVC or CVV). Again, tonal adjustments are fully local to the juncture, showing no impact on earlier tones of the base.

Before we move on, I would like to comment briefly on vowel length, the other distinctive suprasegmental property of Hausa: while morphological tone may be assigned holistically or be merely agglutinative, possibly triggering some local adjustments under strict adjacency, all the cases of length alternation we have observed so far are of a strictly local nature. In fact, this appears to be a general property of the language: while tones may be assigned individually or as entire melodies, there is no assignment of rhythmic length patterns that operates across larger domains, let alone “length harmony”, i.e. the spreading of same length specifications. In essence, length is only

[8] Hausa grades can be roughly thought of as inflectional classes, although some grades (namely 4–7) also encode derivational properties, like totality (4), causative or efferential (5), ventive (6) or medio-passive (7), which makes them resemble the *binyanim* of distantly related Semitic languages: each of the grades is associated with characteristic tone patterns, and an equally characteristic alternation of the final vowel, depending on the presence and nature of the direct object, called *frames*: Frame B is used with pronominal affixes, Frame C with locally realised direct objects, and Frame A elsewhere, including object fronting. See Parsons (1960), Newman (2000) and Jaggar (2001) for in-depth descriptions, as well as Abdoulaye (1992) and Crysmann (2005a) for recent synchronic analyses. Note that, for some grades, the exact tonal pattern changes according to frame. Cf. Table 2 for an overview of affixal and tonal patterns.

ever manipulated locally, while tonal manipulations may operate locally or globally, depending on the morphological construction.

3.3 *Toneless prefixes*

A third type of tonal behaviour triggered by morphological affixation is contributed by prefixation of “toneless” markers, i.e. affixes that are not inherently prespecified for a particular tone: regular and productive formation of pluractionals in Hausa is expressed by prefixation of a CVC reduplicative prefix, where the two consonants are identical to the first root consonant, and the vowel is essentially identical to the first vowel of the root (modulo reduction in closed syllables).

(10) C_1VC_1-

a. dárnàcḗ (H-L-H) – dáddárnàcḗ (H-H-L-H)
'press down/oppress' (GR4 A) (Newman 2000, p. 424)

b. káràntā́ (H-L-H) – kákkáràntā́ (H-H-L-H)
'read' (GR1 A/B) (Newman 2000, p. 424)

c. dā̀gúrā̀ (L-H-L) – dàddā̀gúrā̀ (L-L-H-L)
'gnaw at' (GR2 A) (Newman 2000, p. 425)

d. káràntà (H-L-L) – kákkáràntà (H-H-L-L)
'read' (GR1 C)

e. dā̀gùrí (L-L-H) – dàddā̀gùrí (L-L-L-H)
'gnaw at' (GR2 C)

As illustrated in (10), prefixation of CVC to trisyllabic bases gives rise to quadrisyllabic words. Since melodies for Hausa verbs are maximally tri-tonal, what happens is that the leftmost tone simply spreads to the tonally underspecified prefix, yielding H for H-initial and L for L-initial bases.

Pluractionals of bisyllabic bases display a slightly more intricate pattern: although spreading can still be attested in those paradigms (grades) that are maximally bitonal, such as grades 6 and 7 in (11a–b), or the B and C forms of grade 2 in (11c), grades which witness tri-tonal melodies, such as grades 1 and 4, as well as the A form of grade 2, simply use the tri-tonal melody for pluractionals that we already found with trisyllabic bases in these grades (12). Table 2 provides a synopsis of segmental and tonal patterns for all seven productive grades.

Table 2: Synopsis of Hausa grades

	Frame A[9]		Frame B		Frame C	
Grade	Seg/Len	Tone	Seg/Len	Tone	Seg/Len	Tone
		$\sigma\sigma$ / $\sigma^+\sigma\sigma$		$\sigma\sigma$ / $\sigma^+\sigma\sigma$		$\sigma\sigma$ / $\sigma^+\sigma\sigma$
1	-ā	H-L / H^+-L-H	-ā	H-L / H^+-L-H	-a	H-L / H^+-L-L
2	-ā	L-H / L^+-H-L	-ē	L^+-H	-i	L^+-H
3	-a/-i	L-H / L^+-H-L		–		–
4	-ē	H-L / H^+-L-H	-ē	H-L / H^+-L-H	-ē/-e	H-L / H^+-L-L
5	-r̃	H^+	-shē	H^+	-r̃	H^+
6	-ō	H^+	-ō	H^+	-ō	H^+
7	-u	L^+-H		–		–

(11) a. kā́wṓ (H-H) – kákkā́wṓ (H-H-H)
‘bring’ (GR6) (Newman 2000, p. 424)

b. gyā̀rú (L-H) – gyàggyā̀rú (L-L-H)
‘be well repaired’ (GR7)’ (Newman 2000, p. 424)

c. jḕfḗ (L-H) – jàjjḕfḗ (L-L-H)
‘throw at’ (GR2 B)

(12) a. tā́kā̀ (H-L) – táttā̀kā́ (H-L-H)
‘step on’ (GR1 A) (Newman 2000, p. 424)

b. jḕfā́ (L-H) – jàjjḗfā̀ (L-H-L)
‘throw at’ (GR2 A) (Newman 2000, p. 424)

To summarise, the addition of a syllable by pluractional prefixation incurs a switch of tonal pattern to the appropriate melody associated with trisyllabic words for that paradigm cell. If the melody provides for fewer tones than there are tone-bearing units, automatic spreading applies: occasionally, with trisyllabic pluractionals (depending on the tonal pattern of that grade), and always, across all grades, with quadrisyllabic ones.

Having investigated the basic suprasegmental processes associated with morphological operations in Hausa, namely holistic assignment (“tone-integrating affixes”), tonal affixation (“non-integrating affixes”), and spreading of base tones on to inherently toneless prefixes, we are now in a position to explore, in the following sections,

[9] Segmental shape and tone can both be subject to lexical exception in Frame A, in particular in grade 2.

how these processes can be integrated, in an efficient way, into a computational grammar of Hausa, built on a lean typed feature structure formalism.

4 HAG – A COMPUTATIONAL GRAMMAR OF HAUSA

The treatment of tone discussed in this paper is part of an emerging implemented computational grammar of Hausa, developed in the framework of Head-driven Phrase Structure Grammar. The underlying typed feature logic is a purely conjunctive variant of TDL (=Type Description Language; Krieger 1996), as currently implemented in several processing systems, such as the Linguistic Knowledge Builder (=LKB; Copestake 2002), the PET parser (Callmeier 2000), and the ACE parser and generator (the latter being described in Crysmann and Packard 2012). Owing to its declarative nature, the grammar is fully reversible, i.e. it can be used for both parsing and generation. Furthermore, the symbolic grammar resource is complemented by stochastic models for parse selection and realisation ranking, developed on the basis of the Redwoods treebanking technology (Oepen *et al.* 2002). An overview of the grammar and the major constructions it covers can be found in Crysmann (2012a). The grammar is freely available at `http://svn.emmtee.net/trunk/llf/hag/` under an open-source licence. An online demonstrator of the grammar is hosted at `http://hag.delph-in.net/logon`.

Although the lexicon is still rather small, the grammar already covers a wide range of core constructions of the language. With respect to morphology, the grammar implements inflectional morphology in both the nominal and the verbal domain, including the infamously rich set of plural formation patterns. On the segmental side, morphological rules cover all morphophonological processes attested in the language, including affixation, gemination, as well as partial and full reduplication.

With respect to morphosyntax, the grammar boasts a systematic treatment of direct object marking, the so-called Hausa frames (Parsons 1960). The inflectional approach to Hausa verb frames developed in Crysmann (2005a) has been generalised to nominal categories, including gerunds, prenominal adjectives and possessives (Crysmann 2011), as well as prepositional nouns (Crysmann 2012b).

On the purely syntactic side, the grammar covers local complementation and modification, as well as non-local processes, such as wh-extraction, focus fronting, and relativisation, with both gap and resumptive strategies (Crysmann 2012c, 2015).

Owing to the central status of tone and length for marking morphological and morphosyntactic properties, the grammar has been developed from the ground up to support suprasegmental representations. Particular care has been given to the fact that suprasegmental information is represented to different degrees in textual input: while standard Latin orthography (*Boko*) does not represent tone or length at all, length, but not tone, is marked in the Arabic script (*Ajami*). Scholarly as well as educational work on Hausa, by contrast, tends to fully mark tone and length, although the marking regimes differ: while long vowels are typically marked by macrons, leaving short vowels unmarked, as in the grammars by Newman (2000), Jaggar (2001), and Caron (1991), the Hausa language course by Cowan and Schuh (1976), or the Hausa–French dictionary by Caron and Amfani (1997), there are also clearly alternative marking schemes: Jungraithmayr and Möhlig (1976) use geminated vowel letters to mark length, and Newman and Ma Newman (1977) mark brevity (with a comma below the vowel). In *Ajami*, long vowels are signalled by a combination of letters (ya, wau, alif) and diacritics, distinguishing 5 vowel qualities, whereas short vowels are solely marked by diacritics, distinguishing 3 qualities. As for tone, the most wide-spread convention used in the Hausaist literature is to mark low and falling tone (with grave and circumflex accents), leaving high tone unmarked.

Given this diversity regarding the amount of suprasegmental information being marked, as well as the way it is signalled, a highly flexible approach is called for, if we want to be able to maximally exploit suprasegmental information, if present in the input, while at the same time ensure robustness towards partially marked or unmarked input. Moreover, given the locality of morphological processes regarding segmental material and length specifications as opposed to the potentially non-local assignment of melodies, an autosegmental separation of these pieces of information is inevitable. To this end, the grammar employs a token rewrite system (Adolphs *et al.* 2008) to convert a diacritically marked input string into a featural representation, separating the segmental level from the levels of tone and length specifications. The grammar can be configured at runtime as to which inferences should be drawn from overt suprasegmental specifications in the input: consistent full marking, where suprasegmentally unmarked segments are interpreted as the complement of the tone and length markings found, giving maximal disambiguation, and partial marking, where only overt marking is taken into consideration, permitting sporadic marking of tone or length by the user. While the former is best suited to the processing of edited texts, the latter is intended for interactive input to the grammar, where only critical tones may be marked and strict adherence to a consistent marking regime would appear cumbersome.

Grammar-internally, tone and length are systematically represented at the lexical, morphological and morphosyntactic levels. Thus, on the basis of the interaction of lexical with local and non-local grammatical constraints, suprasegmental information missing from the input can be recovered to a great extent by symbolic means. The residual ambiguity is addressed by means of discriminative parse selection models extracted from a treebank. Regenerating from disambiguated parses provides full reconstruction of tone and length specifications, obeying the full set of constraints imposed by the grammar, both locally and globally. Based on the tight integration of suprasegmental information, the grammar not only reaches a high level of linguistic adequacy, but will ultimately be suitable in a number of application scenarios for which this information is crucial, including text-to-speech synthesis (TTS), and computer-assisted language learning (CALL).

5 TOWARDS AN EFFICIENT AND FLEXIBLE REPRESENTATION OF TONE

5.1 *Tonal tiers in typed feature structures*

The data structures I shall adopt for the representation of suprasegmental information are lists: one list for tone sequences, and another list for vowel length information. Lists already have some intrinsic properties that make them suitable as a representation of tiers: first, in contrast to sets, they permit multiple occurrences of like elements, and, even more importantly, they are ordered, capturing the temporal organisation of the tier. Second, they constitute a much simpler data structure than trees, which again seems to be a desirable property: while there is some evidence suggesting that the distribution of tones may depend on a hierarchical structure, be it morphological or prosodic, there seems to be very little evidence in Hausa as to a hierarchical structure of tonal sequences themselves.[10]

In (typed) feature structure formalisms, lists can be recursively implemented using FIRST/REST or HEAD/TAIL notation: the first element of the list is represented as the value of the HD feature, while the list remainder, itself a list, is represented as the value of TL. As illustrated in Figure 1, the second element will be found under TL.HD, the third element under TL.TL.HD etc.[11] Pure feature structure encoding of lists in HEAD/TAIL notation only directly exposes one end of the data structure, essentially corresponding to a stack or LIFO (= Last In First Out) in terms of data structure: if we recursively build up these lists element by element, we can easily access the last member that has been added to the resulting list, but direct access to the first element ever added will not be straightforward.

$$< \textit{high}, \textit{low}, \textit{low}, \ldots > \; \equiv \begin{bmatrix} \text{HD} & \textit{high} \\ \text{TL} & \begin{bmatrix} \text{HD} & \textit{low} \\ \text{TL} & \begin{bmatrix} \text{HD} & \textit{low} \\ \text{TL} & [\;] \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 1:
HD/TL encoding of lists

Throughout this paper, I shall use the list constructor (|) as an abbreviatory device to partition a list into an initial sequence of elements and a list remainder, a notation familiar from Prolog (Clocksin and Mellish 1981, pp. 52–53), that is also regularly used in HPSG. As for the representation of morphological rules, I assume feature structure descriptions where the entire term represents the properties of the derived structure (the mother in a unary rule), whereas the properties of the morphological daughter, i.e. the base, are embedded under a

[10] Of course, as pointed out by one of the reviewers, trees can be encoded using lists of lists. At the level of tonal tiers, however, appropriateness conditions on list types will ensure that lists cannot be nested. See below in this section on typed lists.

[11] I am using period as a path separator, to avoid confusion with the list constructor (|).

feature DTR (cf. Riehemann 1998; Koenig 1999). Information shared between the mother and the daughter are captured by means of reentrancies (= *token identity* or *structure sharing*), expressed using boxed coreference tags, e.g. $\boxed{t}$ in Figure 2.[12]

Figure 2: Basic list operations

$$\begin{bmatrix} \text{TONE} & < \textit{low} \mid \boxed{t} > \\ \text{DTR} & \begin{bmatrix} \text{TONE} & \boxed{t} \end{bmatrix} \end{bmatrix} \equiv \begin{bmatrix} \text{TONE} & \begin{bmatrix} \text{HD} & \textit{low} \\ \text{TL} & \boxed{t} \end{bmatrix} \\ \text{DTR} & \begin{bmatrix} \text{TONE} & \boxed{t} \end{bmatrix} \end{bmatrix}$$

(a) Add tone (push)

$$\begin{bmatrix} \text{TONE} & \boxed{t} \\ \text{DTR} & \begin{bmatrix} \text{TONE} & < \textit{high} \mid \boxed{t} > \end{bmatrix} \end{bmatrix} \equiv \begin{bmatrix} \text{TONE} & \boxed{t} \\ \text{DTR} & \begin{bmatrix} \text{TONE} & \begin{bmatrix} \text{HD} & \textit{high} \\ \text{TL} & \boxed{t} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

(b) Remove tone (pop)

$$\begin{bmatrix} \text{TONE} & < \textit{low} \mid \boxed{t} > \\ \text{DTR} & \begin{bmatrix} \text{TONE} & < \textit{high} \mid \boxed{t} > \end{bmatrix} \end{bmatrix} \equiv \begin{bmatrix} \text{TONE} & \begin{bmatrix} \text{HD} & \textit{low} \\ \text{TL} & \boxed{t} \end{bmatrix} \\ \text{DTR} & \begin{bmatrix} \text{TONE} & \begin{bmatrix} \text{HD} & \textit{high} \\ \text{TL} & \boxed{t} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

(c) Change tone (pop & push)

Given these rather straightforward assumptions, we are already in a position to capture the kind of operations characteristic of

[12] Lexical rules, as employed here, are description-level rules, as opposed to meta-level rules: i.e. they essentially function like unary branching syntactic rules, with the added functionality of attaching orthographemic changes using a variant of string unification (Calder 1989). Reentrancies between mother and daughter are fully specified in the rule descriptions or the rule types they inherit from. We exclusively rely on the type system to minimise redundancy, in contrast to Meurers (2001), who proposes compilation of a special lexical rule format. While the LKB supports default unification (Lascarides *et al.* 1996; Lascarides and Copestake 1999), neither PET nor ACE do: therefore, the development of HAG is fully monotonic, i.e. devoid of defaults.

tone non-integrating affixation.[13] In essence, there are three basic tonal operations that can be captured by means of morphological rule schemata operating on the tones of the base: addition of a tone (push an element on to the stack), deletion of a tone (pop an element off the stack), and modification, as a combination of push and pop. Figures 2 and 3 illustrate the equivalence (≡) of the simplified list notation with the underlying feature structure representation.

One can add tones to the beginning of the list (Figure 2a), by way of structure sharing the TONE list of the base with the TL of the resulting sign plus a specification of the HD element; one can remove initial elements from the list, by having the resulting sign's TONE list structure shared with the TL of the base only, excluding the HD element of the base (Figure 2b); or one can replace elements (Figure 2c), by combining the two operations.[14] As illustrated in Figure 3a below, while access is not random, it is by no means limited to the first element: any element can be manipulated, as long as it is found at a definite distance from the beginning of the list. Likewise, one can easily define

[13] In contrast to One-Level Phonology (Bird and Klein 1994), I shall assume a weak version of Phonological Compositionality, permitting "feature-changing" operations as the result of the application of morphological rules: while, strictly speaking, any feature value can only ever be made more specific in a monotonic feature logic like the one assumed here, the effect that a constraint imposed on the morphological base may not hold true for a form derived from that base can indeed be captured by assigning different values to the representations of the morphological mother and the daughter, i.e. by not fully equating their phonological representations.

[14] The way tonal modification is presented here, i.e. independent of syllable count or segmental changes is intentional: in the spirit of autosegmental phonology, alterations on one tier may, but need not, be paralleled by according alterations on a different tier. Basic tonal operations, as presented here, are building blocks, which may occur in conjunction with segmental and metrical changes in concrete morphological rules. In Hausa, addition of tones, as represented in the grammar, is usually accompanied by adding a tone-bearing unit (see e.g. suffixation of possessives in Figure 10). Changing a tone need not be, as witnessed by the previous reference marker in Figure 11. Adding or deleting a tone without manipulating the metrical structure would entail shifting of the remainder of the tonal specification. However, Hausa does not seem to provide any clear evidence for this.

Figure 3: Some complex list operations

$$\begin{bmatrix} \textsc{tone} & < \boxed{1}, \textit{low} \mid \boxed{t} > \\ \textsc{dtr} & \begin{bmatrix} \textsc{tone} & < \boxed{1}, \textit{high} \mid \boxed{t} > \end{bmatrix} \end{bmatrix} \equiv \begin{bmatrix} \textsc{tone} & \begin{bmatrix} \textsc{hd} & \boxed{1} \\ \textsc{tl} & \begin{bmatrix} \textsc{hd} & \textit{low} \\ \textsc{tl} & \boxed{t} \end{bmatrix} \end{bmatrix} \\ \textsc{dtr} & \begin{bmatrix} \textsc{tone} & \begin{bmatrix} \textsc{hd} & \boxed{1} \\ \textsc{tl} & \begin{bmatrix} \textsc{hd} & \textit{high} \\ \textsc{tl} & \boxed{t} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

(a) Change second tone

$$\begin{bmatrix} \textsc{tone} & < \boxed{2}, \boxed{1} \mid \boxed{t} > \\ \textsc{dtr} & \begin{bmatrix} \textsc{tone} & < \boxed{1}, \boxed{2} \mid \boxed{t} > \end{bmatrix} \end{bmatrix} \equiv \begin{bmatrix} \textsc{tone} & \begin{bmatrix} \textsc{hd} & \boxed{2} \\ \textsc{tl} & \begin{bmatrix} \textsc{hd} & \boxed{1} \\ \textsc{tl} & \boxed{t} \end{bmatrix} \end{bmatrix} \\ \textsc{dtr} & \begin{bmatrix} \textsc{tone} & \begin{bmatrix} \textsc{hd} & \boxed{1} \\ \textsc{tl} & \begin{bmatrix} \textsc{hd} & \boxed{2} \\ \textsc{tl} & \boxed{t} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

(b) Metathesis

metathesis of a pair of tones (Figure 3b), provided their position is known, which I assume to be the case.[15]

There is one important inherent limitation to lists represented as simple feature structure terms: in lean formalisms without any functional or relational constraints,[16] such as the one we are using here, lists are essentially stacks, i.e. access of any element is straightfor-

[15] In case the relevant elements are found in a finite number of positions, situations can of course be enumerated using multiple rules.

[16] Current alternative implementations of typed feature formalisms include TRALE (Penn 2004), which actually supports relational constraints and disjunction. However, relational constraints provide an additional recursive structure besides the main recursion on the rule backbone. Thus, while lean formalisms will force hidden costs out into the open, such costs can be hidden by relational constraints. However, as we shall see, no purely phonologically motivated recursion steps are needed, other than what is already offered by type expansion. For the purposes of this paper, which aims at assessing the minimal computational power needed to address autosegmental phonology in Hausa, the choice of a purely conjunctive typed feature structure unification formalism without relational constraints must appear preferable, for methodological reasons.

ward from the beginning of the list, but quite hard to determine, in a general fashion, from the end of the list. Thus, in the context of these formalisms, it is of central concern whether tones are represented from left to right or rather from right to left. Given that Hausa is predominantly suffixal, and that spreading also proceeds from right to left, I shall assume for now[17] that the most appropriate encoding of the tonal tier in this language will be from right to left, essentially assuming that tones on the TONE list are represented in inverse order of surface tone, i.e. list-initial tones correspond to rightmost surface tones, whereas list-final tones correspond to leftmost surface tones. Given an encoding in this order, tonal suffixation, including local modifications, as observed for non-integrating affixes, can be straightforwardly captured, using the basic operations illustrated in Figure 2. Sample analyses will be provided in Section 5.3.2.

Having established so far that a right-to-left encoding of tone is most suitable for the treatment of Hausa, facilitating the description of local tonal modifications observed for non-integrating suffixes, we shall now move on to tone-integrating suffixation, including spreading.

An interesting property of typed feature formalisms is that they inherently provide for parameterised list constraints, enabling us to impose some constraint over an arbitrary number of elements. As we shall see shortly, this property is key to our implementation of spreading.

To start with, consider the type hierarchy of basic list types, as given in Figure 4. The supertype *list* is defined to have exactly two immediate subtypes: either an empty list (*e-list*), or a non-empty list (*ne-list*). While the former has no appropriate features, the latter intro-

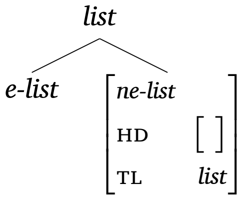


Figure 4:
Basic list type declarations

[17] See, however, Sections 5.2 and 5.4.2 for detailed discussion and a generalised treatment.

duces the features HD and TL. According to the logic of typed feature structures (Carpenter 1992), whenever a feature such as HD is required to be present by some constraint, the type of the feature structure must be at least *ne-list*, and all other constraints associated with this type are enforced. Conversely, once a list has been specialised to *e-list*, it will be incompatible with either HD or TL features, since *e-list* and *ne-list* do not have a common lower bound.

List types and appropriateness, however, will show their full potential once we associate additional properties with a type. More concretely, I shall build on typed list constraints, a powerful, yet efficient way to impose constraints on the members of lists of arbitrary size.[18]

As stated in (13) and (14), one can provide type definitions for tone lists consisting of an arbitrary number of H, or an arbitrary number of L, yielding the type hierarchy in Figure 5. In essence, these list types will provide us with a concise and efficient formalisation of tone spreading.

(13) a. $h^{*}\textit{-list} := \textit{list}.$

b. $h^{*}\textit{-e-list} := h^{*}\textit{-list} \wedge \textit{e-list}.$

c. $h^{*}\textit{-ne-list} := h^{*}\textit{-list} \wedge \textit{ne-list} \wedge \begin{bmatrix} \text{HD} & \textit{high} \\ \text{TL} & h^{*}\textit{-list} \end{bmatrix}$

(14) a. $l^{*}\textit{-list} := \textit{list}.$

b. $l^{*}\textit{-e-list} := l^{*}\textit{-list} \wedge \textit{e-list}.$

c. $l^{*}\textit{-ne-list} := l^{*}\textit{-list} \wedge \textit{ne-list} \wedge \begin{bmatrix} \text{HD} & \textit{low} \\ \text{TL} & l^{*}\textit{-list} \end{bmatrix}$

As defined in (13), the type *h*-list* can be expanded either into *h*-e-list*,[19] a subtype of the empty list type, or else into the non-empty list type, *h*-ne-list*, which restricts the HD element to be *high*. The remainder of the *h*-ne-list* is in turn restricted to be of type *h*-list*, prop-

[18] To the best of my knowledge, list types were first explored in a systematic way by Flickinger (2000). See also Crysmann (2005b) on extended applications of this technique, e.g., for the implementation of type identity.

[19] Throughout this paper, the asterisk on tone specifications (*) is the Kleene star familiar from regular languages, denoting an arbitrary number of repetitions of tone symbols of this type, including zero. This notation should not be confounded with that used for pitch accents in intonation.

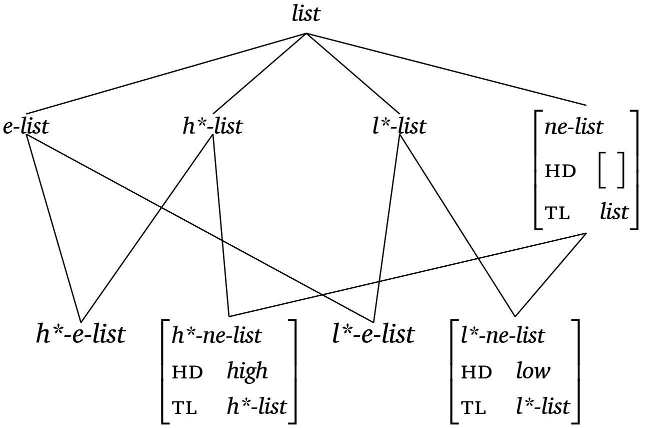


Figure 5: Hierarchy of tone list types

agating the tonal requirement further down the list: if the remainder is empty, nothing much happens, but if it is non-empty, it must be a subtype of both *ne-list* and *h*-list*, resolving it to *h*-ne-list*, fixing the value of the HD element (to *high*), and setting the value of the new list remainder to *h*-list*. Thus, tone list types provide a concise and effective way to push properties across lists of arbitrary length, in essence stating properties independently of the number of list members. More importantly, these constraints are latent, such that expansion is delayed until tonal specifications are accessed by other constraints, with full expansion being reserved to synchronisation with the metrical tier.

Based on this minimal inventory of tone list types, we are now in a position to formalise Hausa tone assignment, including automatic spreading, within the context of typed feature structure grammar. To give a more concrete example, let us consider the case of grade 6 verbs, featuring holistic assignment of a single spreading H tone for both basic verbs and pluractionals. The tonal constraint associated with grade 6 verbs is that of an *h*-list*: depending on the number of tone-bearing units (e.g. two for a basic verb like *kāwō*, and three for the pluractional derived from it), unification of the general *h*-list* constraint with tone lists of appropriate length will yield specialisation of *h*-list* to exactly as many high tones as required.

In order to understand the exact workings of list type constraints, let us briefly work through the example in Figure 6a: unification of *h*-list* with the outer level of the two-element list enforces specialisation of *h*-list* to *h*-ne-list*, entailing specialisation of the top-level HD

Figure 6: Automatic expansion of tone list constraints

$$\textit{h}^*\textit{-list} \wedge \begin{bmatrix} \textit{ne-list} & \\ \text{HD} & [\,] \\ \text{TL} & \begin{bmatrix} \textit{ne-list} & \\ \text{HD} & [\,] \\ \text{TL} & \textit{e-list} \end{bmatrix} \end{bmatrix} \equiv \begin{bmatrix} \textit{h}^*\textit{-ne-list} & \\ \text{HD} & \textit{high} \\ \text{TL} & \begin{bmatrix} \textit{h}^*\textit{-ne-list} & \\ \text{HD} & \textit{high} \\ \text{TL} & \textit{h}^*\textit{-e-list} \end{bmatrix} \end{bmatrix}$$

(a) Bisyllabic *kā́wṓ*

$$\textit{h}^*\textit{-list} \wedge \begin{bmatrix} \textit{ne-list} & \\ \text{HD} & [\,] \\ \text{TL} & \begin{bmatrix} \textit{ne-list} & \\ \text{HD} & [\,] \\ \text{TL} & \begin{bmatrix} \textit{ne-list} & \\ \text{HD} & [\,] \\ \text{TL} & \textit{e-list} \end{bmatrix} \end{bmatrix} \end{bmatrix} \equiv \begin{bmatrix} \textit{h}^*\textit{-ne-list} & \\ \text{HD} & \textit{high} \\ \text{TL} & \begin{bmatrix} \textit{h}^*\textit{-ne-list} & \\ \text{HD} & \textit{high} \\ \text{TL} & \begin{bmatrix} \textit{h}^*\textit{-ne-list} & \\ \text{HD} & \textit{high} \\ \text{TL} & \textit{h}^*\textit{-e-list} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

(b) Trisyllabic *kákkā́wṓ*

value to *high* and of TL to *h*-list*. This specialisation of the TL value will in turn trigger unification with the value *ne-list* which yields the most general subtype of *h*-list* and *ne-list*, namely *h*-ne-list* of Figure 5. The constraint associated with this latter type will again be applied to the feature structure under TL, specialising TL.HD to *high*, and propagating the tone list type on to TL.TL. Unification with the empty list under TL.TL finally resolves to *h*-e-list*.

Before moving on, let us briefly discuss how the suprasegmental representations are synchronised. As established at the end of Section 3.2, length specifications are only ever modified locally and, more crucially, do not require underspecification, in contrast to tonal spreading, for example. This makes the syllable or length tier an ideal timing tier. By contrast, holistic assignment and tonal spreading favour an underspecified description. During morphological construction, we therefore state constraints using underspecified descriptions, treating representations of tone separate from length. At the level of the maximal morphological word, however, we use the length of the LEN list, our timing tier, to determine the exact length of the TONE list, effectively expanding list constraints to exactly the number of tones required.[20] The delayed synchronisation of tone lists with the timing tier is not only clearly in the spirit of an autosegmental approach, but it also provides a highly efficient implementation, avoiding repeated recursion over autosegmental tiers. Furthermore, given the monotonic nature of HPSG's underlying feature structure formalism, a single, delayed evaluation is fully sufficient.

5.2 *Direction of tone assignment*

We have seen so far that the tonal effects of morphological processes in Hausa assign a privileged status to the right edge: tone non-integrating affixes simply add tones on the right, potentially supplanting or modifying a base-final tone, whereas melodies assigned by tone-integrating affixes are typically assigned one by one, starting with the suffix, with the leftmost tone spreading leftwards on to any preceding syllables. In fact, as suggested by Newman (1986, 2000) and Jaggar (2001), the standard direction of association in Hausa is from right to left. While this view may contradict the conventions proposed by Clements and Ford (1979), who suggest a universal left-to-right spreading convention, right-to-left association and spreading nevertheless appear to be particularly well motivated by the empirical patterns of the language.

To start with, tone patterns assigned to lexical bases are probably somewhat inconclusive, as argued by Schuh (1989, p. 257). He discusses abstract L H sequences of quadrisyllabic lexical nouns and observes that all three logically possible patterns are attested in the Hausa lexicon: L-H-H-H (bùlā́gúr̃ṓ 'trip'), L-L-H-H (*gwàlàmníyā́* 'speaking unitelligibly'), and L-L-L-H (*ànnàshùwā́* 'feeling happy').

The situation, however, becomes much clearer once we consider grammatically assigned tone patterns: out of the 15 plural patterns

[20] Technically, synchronisation of TONE and LEN lists is performed by unification: in order to keep type hierarchies of tone and length values distinct, we actually maintain a "shadow" list (--LEN) as part of our LEN definition, that will have the same length as the main list, yet does not constrain the type of the list elements, as shown below.

i. *len-list* := *list* ∧ [--LEN *list*]

ii. *len-e-list* := *len-list* ∧ *e-list*

iii. *len-ne-list* := *len-list* ∧ *ne-list* ∧ [--LEN < [ ] | $\boxed{1}$ > , TL.--LEN $\boxed{1}$]

identified by Newman (2000) (cf. Table 1), 7 have an initial tonal plateau and final alternation of H and L (classes II, III, VI, VII, IX, XI, XIII), whereas only 1 class observes the opposite pattern (class VIII), showing a fixed H-L-H-H melody. However, since the plurals in this class are all quadrisyllabic, no really strong case can be made for spreading of a final H. The remaining plural tone patterns are inconclusive, since they are either monotonal (all plateau), or show no evidence of spreading (no plateau). Furthermore, the right-to-left perspective on assignment neatly aligns with the fact that any unambiguous examples of tone-integrating affixes are suffixal.[21] Considering the tonal patterns within the verbal grade system (cf. Table 2), we find this pattern confirmed: 8 out of 10 minimally bi-tonal patterns show alternating tones on the right, while having a tonal plateau, a potential indicator of spreading, on the left. Furthermore, the two somewhat exceptional H-L-L patterns do feature spreading on the left with quadrisyllabic verbs. Moreover, the point of transition between H and L is always at a fixed distance when counting from the right, yet would be variable when proceeding from the left.

Another piece of evidence in favour of an alignment between suffixation and tone assignment comes from multiple suffixation of tone-integrating affixes: as discussed in Newman (1986), regular Hausa past participles are formed by affixation of a tone-integrating reduplicative *-aCC* suffix, where C reduplicates the last consonant of the base, and an agreement marker (*-ē,-iyā,-ū*). In the singular, the base is characterised by an LH tone pattern, whereas the masculine and feminine agreement markers *-ē* and *-iyā* carry high tone. In the plural, however, we observe a final H on the agreement marker *-ū*, preceded by an all-L tone sequence.

[21] There are two not fully productive derivational patterns featuring both prefixation and holistic tone assignment, namely agentive nouns and a subclass of ethnonyms. However, since both patterns are circumfixal, i.e. they simultaneously involve suffixation, they do not provide conclusive counter-evidence to the otherwise systematic observation that tone-integrating affixes are suffixal. Moreover, as witnessed by the contrast between masculine and feminine agentive nouns, choice of tonal pattern is most likely associated with the suffixal part of the circumfix, rather than with the segmentally and tonally constant prefix. I will discuss these in full in Section 5.4.2, including a generalised formal analysis of tone spreading and prespecification.

masculine	*feminine*	*plural*	*gloss*
dàfáffḗ	dàfáffíyā́	dàfàffū́	‘cooked’
gā̀gàrárrḗ	gā̀gàrárríyā́	gā̀gàràrrū́	‘rebellious’
yàgàlgàlállḗ	yàgàlgàlállíyā́	yàgàlgàlàllū́	‘torn into pieces’

Table 3:
Hausa past participles

Newman (1986) explains these patterns by means of the interaction of two tone-integrating suffixes: a tone-integrating LH participle marker -*aCC*, and a tone-integrating LH plural marker -*ū*, which is independently attested in noun class IX, with identical segmental and suprasegmental properties. While non-integrating singular agreement affixes -*ē* and -*iyā* leave the LH participial tone pattern intact, suffixation of the tone-integrating plural marker -*ū* replaces the characteristic participial tone pattern with that of the plural marker, again applied from right to left.

Leben (1978) has suggested unifying the case of Hausa's apparent preference for right-to-left assignment and spreading with the left-to-right convention (Clements and Ford 1979) standardly assumed within autosegmental phonology at the time. He reanalyses the Hausa participle facts, using a combination of lexical/morphological prespecification and automatic left-to-right spreading of an initial floating H: taking the example of Hausa past participle formation in Table 3 above, the tonal pattern of a form, such as *gā̀gàrárrḗ*, is analysed as having a pre-linked tone on the past participle marker -*aCC*, preceded by a floating H that associates (and spreads) from left to right. Similarly, the pre-linked L will spread to the end of the word. While this may work particularly well for the case at hand, given the identical final tones, any explicit account of Hausa lexical tone assignment and tone-integrating affixes will still have to establish the exact location of pre-linked tones, which are found in this case at a fixed distance from the edge only when counting from the right (cf., again the examples in Table 3). Thus, even if there is a credible analysis of spreading from the left, pre-linking still needs to proceed relative to the right edge. Finally, any approach that draws on pre-linking requires a non-monotonic logic, since association of unassociated tones has to check first whether or not a tone has already been assigned, which is not necessary when using tone list constraints, as proposed here, which are assigned in a fully monotonic fashion.

There is, however, additional evidence for connecting spreading to a standard right-to-left association in Hausa: while we do find unambiguous examples of toneless *prefixes* in the language (see Section 3.3), there is no such unambiguous evidence for toneless *suffixes*. In contrast to pluractional prefixes, which display an alternation of tone depending on that of the following tone, cases of toneless suffixes are mostly non-existent, or inconclusive, like the past participle agreement markers discussed above, where the tonal specification is invariant (always H in this case).

Another piece of evidence cited by Leben (1978) in favour of left-to-right association comes from vowel epenthesis:[22] the Hausa lexicon has a rather small number of words that are consonant-final. Among these, a subset has an alternate form, the use of which becomes obligatory, for phonotactic reasons, in combination with the linker *-n/-r* (see Section 3.2 above).

(15) HL-final bases

a. fâm – fámī̀
'pound'

b. àlhàmîs – àlhàmī́shī̀
'Thursday' (Leben 1978, p. 207)

(16) H-final bases

a. bā̀búr – bā̀búrī́
'motorbike'

b. àlján – àljánī́
'imp' (Leben 1978, p. 207)

(17) L-final bases

a. mālàm – mālàmī
'teacher'

b. fénsìr – fénsìrī
'pencil' (Leben 1978, p. 207)

In order to provide a unified account of the final tone observed in (15)–(17), Leben (1978) suggested that the epenthetic vowel itself is

[22] Thanks to one of the anonymous reviewers for pointing this out as potential evidence favouring left-to-right association.

toneless, which immediately accounts for the distribution of the falling contour tone over the last two tone-bearing units in (15), and using left-to-right spreading, for the tonal identity of the epenthetic vowel to the preceding syllable in (16). As for L-final bases in (17), he suggested spreading of the final L, paired with subsequent application of Low Tone Raising (Leben 1971), a hypothesised productive rule of Hausa that raises a low on heavy (CVC or CVV) final syllables when preceded by a low tone.

However, there are several reasons to question the validity of this analysis: first, epenthesis hardly enjoys the status of a productive rule of the language. According to Newman (2000, p. 307), the majority of consonant-final bases (over 250) in Hausa do not give rise to epenthesis (e.g. *mā̀shîn* ‘motor cycle’), but rather use syntactic means to encode, e.g. possessive and previous reference marking. Second, the segmental make-up of the epenthetic form is not fully predictable from that of the consonant-final form, whereas the short form can be predicted from the long form: variation includes the quality of the epenthetic vowel, which is mostly *ī*, but sometimes *ū* (18), the quality of the final consonant (19), and the length of the penult, which is mostly long, but sometimes short (20).

(18) hàrâm – hàrā́mū̀
‘unlawful’ (Newman 2000, p. 307)

(19) ràsît – ràsī́ɗī̀
‘receipt’ (Newman 2000, p. 307)

(20) mùtûm – mùtúmī̀
‘man’ (Newman 2000, p. 307)

Newman (2000) therefore reanalyses the final vowel as a latent one, essentially proposing clipping rather than epenthesis, which enables him to account in a straightforward way for the limited productivity, as well as their segmental, metrical, and tonal properties: all of these can be derived on the basis of general, undisputed, fully regular phonotactic properties of the language, invoking general restrictions on coda segments (consonants), syllable weight (vowel shortening in closed syllables), and tone (simplification of LH).

Finally, and most importantly, the status of Low Tone Raising as a (synchronic) phonological rule of Hausa in itself is not unproblematic:

as discussed in Newman and Jaggar (1989), systematic and sporadic exceptions to this rule can be observed throughout the grammar of Hausa: they cite seven phenomena in total (see Schuh 1989, for arguments discarding lengthening in questions as intonational in nature) where this rule is indisputably violated at the surface, including regular plural formation of augmentative adjectives, as illustrated in (26), some imperatives, ideophonic adjectives, adverbs, and action nouns, and recent loans from English. Furthermore, they argued that several putative applications of this rule necessitate unorthodox assumptions regarding word boundaries. Schuh (1989), in a reply to Newman and Jaggar, argued that the generalisation expressed by Low Tone Raising can be saved, once the conditions are suitably refined: he suggested in particular that Low Tone Raising may only apply to sequences of singly associated tones, not to spreading of a single L.[23] However, this specific refinement will not help in the case at hand, since Leben's analysis crucially depends on the combination of spreading and Low Tone Raising in order to derive the surface patterns in (17), in particular, since he explicitly argued in his 1978 paper in favour of multiple association over copying.

To summarise, lexical and morphological assignment of tone in Hausa strongly militates for a right-to-left regime. With spreading, both directions remain as an option, with right-to-left spreading keeping an edge over its competitor, both in terms of a better match between association and spreading, and the empirical asymmetry regarding the privileged existence of toneless prefixes vs. suffixes. The idea of language-specific directions of association and spreading may run counter to universalist ideas about uniformity: faced with the empirical evidence in Hausa, however, it should appear as equally odd to enforce a universalist left-to-right view of spreading, while still maintaining the opposite picture for lexical and morphological tone assignment. I shall therefore conclude that the analysis advanced by Newman (1986, 2000) still remains valid.

Within the context of the current formal approach couched in terms of tone list constraints, prevalence of a single, albeit language-

[23] Despite this qualification, Schuh (1989) equally rejected the status of Low Tone Raising as a phonological *rule* of Hausa, picturing it rather as a lexical constraint of the language.

specific regime is actually a welcome result: given that a stack-like typed list encoding of the tonal tier like the one proposed here confines augmentative, modifying, and subtractive operations to one end of the list, and spreading to the other, we shall give preference to any analysis that treats assignment and association in a symmetrical way. Moreover, a representation encoding both lexical and morphological holistic assignment in the same direction (from right to left) not only facilitates the implementation of spreading by means of list types, it is also beneficial to the treatment of agglutinative tone in a suffixing language such as Hausa, essentially exposing the rightmost tone(s) as the top of the stack, directly available for modification and addition.

5.3 *Suffixes and tone*

Now that we have established a preferential direction for tone assignment and spreading, I shall show how the phenomena we have considered so far can be represented in the context of a Head-driven Phrase Structure Grammar (HPSG) of Hausa, crucially building on the aforementioned typed tone lists.

5.3.1 Tone-integrating affixes

The first type of morphological rules I shall discuss pertains to tone-integrating suffixes, i.e. holistic assignment of melodies. Throughout this section I shall leave out the description of morphosyntactic and purely morphological properties, focusing on segmental and suprasegmental changes instead.[24] As depicted in Figure 7, the suprasegmental effects of regular *-ōCī* suffixation are captured by means of constraints on the TONE and LEN lists, which are both encoded from right-to-left: regarding vowel length, the rule ignores the last length specification of the base and adds two specifications for long vowels. Length specifications for syllables other than the last are shared by the morphological mother ($\boxed{l}$). The segmental changes induced by the rule are

[24] For ease of exposition, I shall employ a feature structure encoding of the segmental representation, rather than the string substitution patterns that are used in the implemented grammar, which are conceived as a variant of string unification (Calder 1989). See Copestake (2002) for details of the orthographemic machinery used in DELPH-IN (`http://www.delph-in.net`) grammars and processing platforms.

represented schematically on the SEG list: the final vowel of the base is suppressed and the last root consonant is reduplicated ($\boxed{c}$), with the vowels *o* and *i* interspersed.

Figure 7:
Morphological rule for
noun class I plural formation

$$\left[\begin{array}{ll}
\textsc{seg} & \boxed{s} \oplus \left\langle \boxed{c}, \mathrm{o}, \boxed{c}, \mathrm{i} \right\rangle \\
\textsc{supra} & \left[\begin{array}{ll} \textsc{tone} & \textit{h*-list} \\ \textsc{len} & \left\langle \textit{long}, \textit{long} \mid \boxed{l} \right\rangle \end{array}\right] \\
\textsc{dtr} & \left[\begin{array}{ll} \textsc{seg} & \boxed{s} \oplus \left\langle \boxed{c}\ \mathrm{C}, \mathrm{V} \right\rangle \\ \textsc{supra} & \left[\begin{array}{ll} \textsc{tone} & \textit{list} \\ \textsc{len} & \left\langle \left[\ \right] \mid \boxed{l} \right\rangle \end{array}\right] \end{array}\right]
\end{array}\right]$$

The assignment of an all-H melody to the morphological complex, however, directly makes use of the typed tone lists introduced in Section 5.1 above. Note moreover that, with tone-integrating suffixes, which indiscriminately ignore the tonal specification of the base, the tonal description of the base is highly underspecified.

Class II suffixation is a variation on the same theme (Figure 8): the base-final vowel is suppressed and replaced with *-ai*. Accordingly, the length specification of the final syllable is constrained to be short: again, this is modelled by suppressing the length specification of the final syllable of the base, together with the addition of a *short* element to that of the morphological complex.

Figure 8:
Morphological rule for noun class II
plural formation

$$\left[\begin{array}{ll}
\textsc{seg} & \boxed{s} \oplus \left\langle \mathrm{a}, \mathrm{i} \right\rangle \\
\textsc{supra} & \left[\begin{array}{ll} \textsc{tone} & \left\langle \textit{high} \mid \textit{l*-list} \right\rangle \\ \textsc{len} & \left\langle \textit{short} \mid \boxed{l} \right\rangle \end{array}\right] \\
\textsc{dtr} & \left[\begin{array}{ll} \textsc{seg} & \boxed{s} \oplus \left\langle \mathrm{V} \right\rangle \\ \textsc{supra} & \left[\begin{array}{ll} \textsc{tone} & \textit{list} \\ \textsc{len} & \left\langle \left[\ \right] \mid \boxed{l} \right\rangle \end{array}\right] \end{array}\right]
\end{array}\right]$$

The L^{+}-H melody is again assigned independently of the tonal make-up of the base. Instead of employing a tone-list constraint for the entire list, we specify non-spreading tones at the beginning of the list (here: *high*) followed by a tone-list constraint on the remainder (*l*-list*).

5.3.2 Non-integrating suffixes

The second major type of morphological rules pertains to tone non-integrating affixes. A trivial case is that of consonantal suffixes, which do not add any tonal specification at all, as witnessed by the linker in Figure 9: while a final *-r* is added to the list of segments, and the length specification of the final syllable is adjusted to *short*,[25] the tonal specification of the base is merely passed on in its entirety ($\boxed{t}$).

$$\begin{bmatrix} \textsc{seg} & \boxed{s} \oplus \langle \text{r} \rangle \\ \textsc{supra} & \begin{bmatrix} \textsc{tone} & \boxed{t} \\ \textsc{len} & \langle \textit{short} \mid \boxed{l} \rangle \end{bmatrix} \\ \textsc{dtr} & \begin{bmatrix} \textsc{seg} & \boxed{s} \\ \textsc{supra} & \begin{bmatrix} \textsc{tone} & \boxed{t} \\ \textsc{len} & \langle [\;] \mid \boxed{l} \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 9:
Morphological rule for "genitive" linker *-r*

A slightly more interesting example is contributed by the possessive pronominal affix, as shown in Figure 10: here, the tone of the suffix, and its length specification are added to the respective suprasegmental lists, with the TONE ($\boxed{t}$) and LEN value ($\boxed{l}$) of the base being identified only with the list remainder.

$$\begin{bmatrix} \textsc{seg} & \boxed{s} \oplus \langle \text{ta} \rangle \\ \textsc{supra} & \begin{bmatrix} \textsc{tone} & \langle \textit{low} \mid \boxed{t} \rangle \\ \textsc{len} & \langle \textit{short} \mid \boxed{l} \rangle \end{bmatrix} \\ \textsc{dtr} & \begin{bmatrix} \textsc{seg} & \boxed{s} \\ \textsc{supra} & \begin{bmatrix} \textsc{tone} & \boxed{t} \\ \textsc{len} & \boxed{l} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

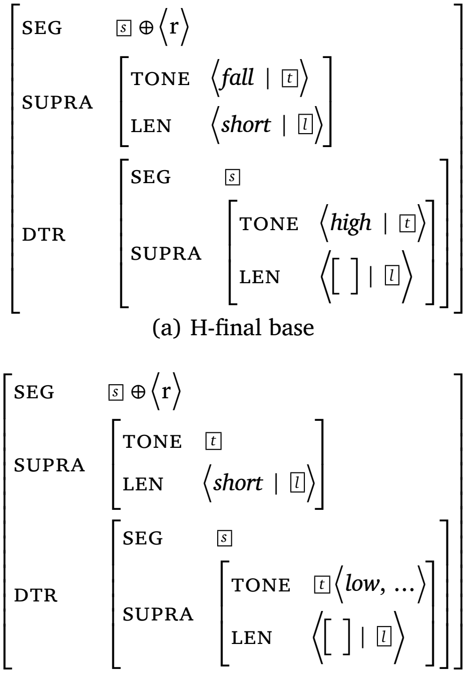
Figure 10:
Morphological rule for possessive pronominal suffix *-tà*

The final example of a non-integrating tone pertains to the previous reference marker *-r̀/-ǹ*: in terms of segmental information and

[25] Recall that there are no long vowels in Hausa closed syllables

length specifications, the rule in Figure 11 is identical to that for the linker in Figure 9, for obvious reasons. Tonally, however, this rule is clearly distinct: while for L-final bases the tonal specification of the base is carried along unaltered, affixation of the previous reference marker changes a final H to a fall.

Figure 11: Morphological rules for previous reference marker -r̀



(a) H-final base

(b) L-final base

Having presented morphological rules for both tone-integrating and tone non-integrating affixation, we are in a position to illustrate how these rules interact. As an example, let us consider the possessive of a class II plural noun, like *tambayōyîn* ‘the aforementioned questions’. By way of illustration, we shall embed the tone-integrating plural rule inside the non-integrating rule for the previous reference marker, as shown in Figure 12.

Starting with the outer rule of previous reference marking, we see that it constrains the final tone of the morphological complex to be *fall* (under SUPRA.TONE.HD), just in case the base is H-final (under DTR.SUPRA.TONE.HD). Similarly, it constrains the final length under

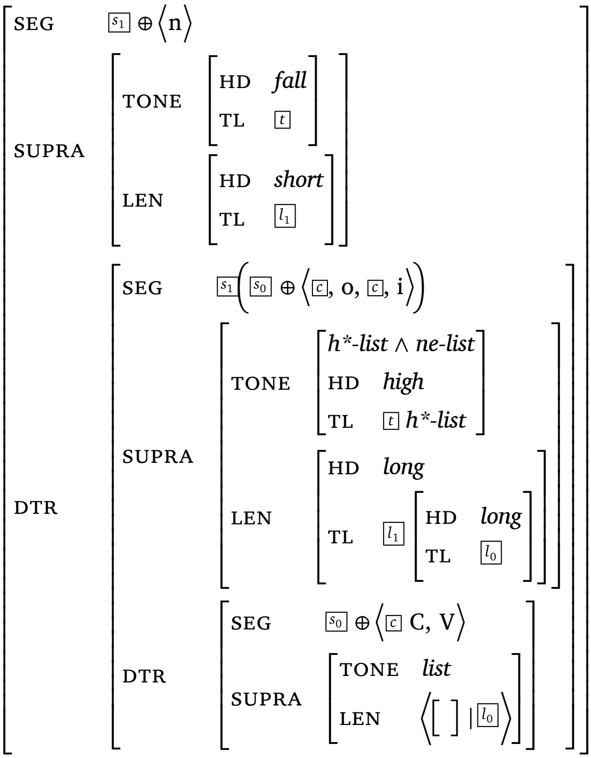


Figure 12:
Interaction between tone-integrating and non-integrating affixation

SUPRA.LEN.HD to *short*, a specification that replaces the final length specification of the base (cf. Figure 11 for the rule in isolation). The inner rule of class I plural formation constrains its mother to be all-H: cf. the constraint *h*-list* under DTR.SUPRA.TONE. Given that the outer rule references the first element (and, for that matter, the list remainder) on this underspecified *h*-list*, the list type is coerced into the type *ne-list*. Given our hierarchy of list types in Figure 5, the most general subtype that is both of type *h*-list* and *ne-list* is *h*-ne-list*. Thus, the conjunction of two types will automatically resolve to the most general subtype of the two, if defined, or otherwise yield a unification failure. Resolution to the subtype, however, automatically enforces any further constraints associated with this type, such as the constraint that the HD value be *high* and the TL value a list of type *h*-list*, enforcing the constraint on DTR.SUPRA.TONE.TL, in our example. Since the constraint on the expanded type *h*-ne-list* is compatible with that of the outer rule, rule application succeeds and the underspecified spreading list constraint is pushed one element down. Let us suppose that we had tried to apply the outer rule to a base specified as *l*-list*: in this case, a constraint restricting the first element of this list to *high* will equally trigger the expansion of *l*-list* to *l*-ne-list*. However, the value of HD imposed by the outer rule (*high*) will fail to unify with that of the type constraint (*low*), blocking application of the rule in this case, as desired. Thus, underspecified list constraints expand, as required, whenever any of their members are accessed. As a result, constraints on tonal identity are virtually present, without our having to keep track of the number of instances they may apply to, ultimately providing us with a very general and efficient approach.

5.4 *Prefixes and tone*

Having discussed the two major modes of operation for suffixation, we shall now turn to the more restricted cases of prefixation. First, I shall discuss how the present approach to spreading naturally extends to toneless prefixes, and then address the phenomenon of tonally prespecified prefixes that I have glossed over in the discussion so far. Finally, I shall generalise the present approach in such a way as to permit morphological operations on tone with both prefixation and suffixation, and show how this integrates with our approach to spreading.

5.4.1 Toneless prefixes

I have argued in the previous section that the overwhelmingly suffixal nature of Hausa, both segmentally and suprasegmentally, favours a representation of tone and length that facilitates access on the right, and I have therefore suggested encoding both TONE and LEN lists from right to left. While this is certainly a reasonable decision, we still need to provide a solution for the few cases of prefixation that nevertheless exist in the language.

One such instance of prefixation was observed in Section 3.3: pluractionals in Hausa are formed by prefixation of a segmentally underspecified reduplicative syllable (*CVC-*). From a tonal perspective, we observed that pluractionals constitute another case of holistic assignment of melodies, including spreading. While prefixation of segmental material poses no problem for the formalism in use (the implementation of string unification provides both prefix and suffix constructs), this is not the case for suprasegmental information, such as vowel length, which is represented using a feature structure encoding of lists. As stated in Section 5.1, in lean typed feature formalisms without relational constraints, arbitrary manipulations are easy at the beginning of the list, yet harder at the end of lists of indeterminate or arbitrary length. If, however, we only need to add elements to the end of a list, unification provides a solution: *difference lists* (familiar from Prolog, for example, Clocksin and Mellish 1981, Chapter 3.8) extend the functionality of ordinary lists by maintaining a pointer to the (open) end of the list, represented here as the feature LAST. As illustrated in Figure 13, concatenation of two lists then proceeds by unifying the LAST feature of the first difference list with the LIST feature of the second list. Since the LAST feature of the first difference list is token identical with the list remainder of that difference list's LIST value, the second list will just wind up at the end of the first. In order to facilitate further list concatenation, the LAST feature of the newly formed difference list will be identical to the LAST feature of the second list.

$$\begin{bmatrix}\textsc{list} & \left\langle \textit{long, long} \mid \boxed{1}\right\rangle \\ \textsc{last} & \boxed{1}\end{bmatrix} + \begin{bmatrix}\textsc{list} & \boxed{1}\left\langle \textit{short} \mid \boxed{2}\right\rangle \\ \textsc{last} & \boxed{2}\end{bmatrix}$$

$$= \begin{bmatrix}\textsc{list} & \left\langle \textit{long, long, short} \mid \boxed{2}\right\rangle \\ \textsc{last} & \boxed{2}\end{bmatrix}$$

Figure 13:
List concatenation using difference lists

How is this applied now to the task at hand? In order to integrate length prefixation by means of difference lists, the first thing we have to do is to provide a list representation for the suprasegmental prefix information, and a principle that ensures concatenation using unification. To this end, I shall introduce the feature PREFIX which takes as its value a suprasegmental structure consisting of TONE and LEN difference lists. In a strictly analogous fashion, I shall generalise the TONE and LEN lists under SUPRA to be difference lists. As depicted in Figure 14, concatenation of the prefixal tone and length lists can be effected by a principle that identifies the end of the suprasegmental lists of the base with the beginning of the prefixal lists. Since we still want to be able to modify, add, or delete tone and length specifications at the beginning of the suprasegmental list, we do not constrain the beginning of the lists under SUPRA to be identical to the beginning of the lists under DTR, but rather leave this to the individual morphological rules.

Figure 14: Concatenation of DTR and PREFIX difference lists

$$\begin{bmatrix} \text{SUPRA} & \begin{bmatrix} \text{TONE} & \begin{bmatrix}\text{LIST} & \textit{list}\\ \text{LAST} & \boxed{t_3}\end{bmatrix}\\ \text{LEN} & \begin{bmatrix}\text{LIST} & \textit{list}\\ \text{LAST} & \boxed{l_3}\end{bmatrix}\\ \text{PREFIX} & \begin{bmatrix}\text{TONE} & \begin{bmatrix}\text{LIST} & \boxed{t_2}\\ \text{LAST} & \boxed{t_3}\end{bmatrix}\\ \text{LEN} & \begin{bmatrix}\text{LIST} & \boxed{l_2}\\ \text{LAST} & \boxed{l_3}\end{bmatrix}\end{bmatrix}\end{bmatrix}\\ \text{DTR} & \begin{bmatrix}\text{SUPRA} & \begin{bmatrix}\text{TONE} & \begin{bmatrix}\text{LIST} & \textit{list}\\ \text{LAST} & \boxed{t_2}\end{bmatrix}\\ \text{LEN} & \begin{bmatrix}\text{LIST} & \textit{list}\\ \text{LAST} & \boxed{l_2}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

Let us now consider the morphological rule for pluractionals, shown in Figure 15: apart from the prefixation of segmental material (on SEG), the rule specifies a prefixal length specification for a *short* vowel under SUPRA.PREFIX.LEN, as well as an underspecified prefixal tone.[26]

According to the principle in Figure 14, the lists of prefix tone and syllable specifications under SUPRA.PREFIX will be appended to the lists under DTR.SUPRA, and the LAST value of the resulting lists on SUPRA will be set to the LAST values on the PREFIX lists. Since pluractional prefixation does not involve any changes at the right end of the base and, more importantly, no suprasegmental ones, we equate the beginning of the lists on SUPRA with those on BASE.

[26] This is mainly for symmetry: since tone assignment is holistic, using open list constraints, and TONE lists are synchronised with LEN, we might just as well have specified an empty difference list here.

$$
\left[\begin{array}{ll}
\textsc{seg} & \left\langle \boxed{c}, \boxed{v}, \boxed{c}, \boxed{c}, \boxed{v} \right\rangle \oplus \boxed{s} \\
\textsc{supra} & \left[\begin{array}{ll}
\textsc{tone} & \left[\textsc{list}\ \boxed{t}\right] \\
\textsc{len} & \left[\textsc{list}\ \boxed{l}\right] \\
\textsc{prefix} & \left[\begin{array}{ll}
\textsc{len} & \left[\begin{array}{ll}
\textsc{list} & \left\langle \mathit{short} \mid \boxed{l} \right\rangle \\
\textsc{last} & \boxed{l}
\end{array}\right] \\
\textsc{tone} & \left[\begin{array}{ll}
\textsc{list} & \left\langle \left[\ \right] \mid \boxed{e} \right\rangle \\
\textsc{last} & \boxed{e}
\end{array}\right]
\end{array}\right]
\end{array}\right] \\
\textsc{dtr} & \left[\begin{array}{ll}
\textsc{seg} & \left\langle \boxed{c}\ C, \boxed{v}\ V \right\rangle \oplus \boxed{s} \\
\textsc{supra} & \left[\begin{array}{ll}
\textsc{tone} & \left[\textsc{list}\ \boxed{t}\right] \\
\textsc{len} & \left[\textsc{list}\ \boxed{l}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]
$$

Figure 15:
Pluractional rule for trisyllabic bases

Holistic assignment of tone melodies to verbs depends essentially on the inflection class (grade), the particular paradigm cell (frame in Hausaist terminology) and the syllable count. Since prefixal syllables will wind up on the SUPRA.LEN list of pluractionals, we can select the appropriate melody, just as we would do with ordinary bisyllabic or trisyllabic verbs. Furthermore, given bijection between tone and length specifications at the word level, tone-list constraints will be expanded to exactly the right number of syllables, triggering spreading whenever the number of length (=syllable) specifications surpasses the number of tones.[27]

5.4.2 Tone-specified prefixes

The second type of prefixation we are going to investigate concerns tone-specified prefixes, as witnessed by two derivational processes of the language: agentive nouns and ethnonyms.[28]

[27] In a purely technical sense, the syllable count can of course never surpass the number of tones, owing to the fact that we consistently use tone-list constraints, which can denote lists of arbitrary length. But, for the purposes of clarity, I shall continue to use the current wording.

[28] I have deferred discussion of these data to this point, essentially for expository purposes: as we shall see shortly, an integrated account of prespecified prefixal tone with spreading calls for a revision of our treatment of spreading that would otherwise have been difficult to grasp.

Agentive nouns in Hausa are derived by prefixation of a high short syllable *ma*, and a gender/number marking suffix, which is *-ē* for masculine singular, *-ìyā* for feminine singular, and *-ā* for plural. The prefixal *ma-* can also be found in other deverbal nominalisations, like instrumentals (*mabū̀ɗī* ‘key/opener’) and locatives (*maƙērā* ‘smithy’), the latter of which, however, differ from agentives not only in the choice of the suffix, but also in terms of their tonal specification, carrying an all-H melody. Amongst the agentive nominalisations, the masculine singular and plural forms deserve particular attention: as described e.g. by Newman (2000) and Jaggar (2001), these forms are characterised by an LH melody, with the initial L spreading to the left, up to, but excluding the prefixal high *ma-*. Note that, again, we are dealing with tone-integrating affixes here, indiscriminately replacing the lexical tone specification with a holistic melody.

(21) a. húkùntā́ ‘to judge’ – máhùkùncī́ ‘judge (M)’ –
máhùkùntā́ ‘judges’ (Newman 2000, p. 52)

b. tsṑrátà ‘be afraid’ – mátsṑràcī́ ‘a coward (M)’ –
mátsṑràtā́ ‘cowards’ (Newman 2000, p. 52)

While it is difficult to find examples of agentive nouns derived from quadrisyllabic bases, a spreading analysis of the LLH pattern observed with trisyllabic bases can instead be motivated by the tonal properties of agentive nouns with bisyllabic or monosyllabic bases.

(22) a. ƙḗrā̀ ‘to forge’ – máƙḕrī́ ‘smith (M)’ –
máƙḕrā́ ‘smiths’ (Jaggar 2001, p. 108/p. 13)

b. sṓ ‘to want’ – másṑyī́ ‘lover (M)’ –
másṑyā́ ‘lovers’ (Jaggar 2001, p. 108)

c. shā́ ‘to drink’ – máshā̀yī́ ‘drinker (M)’ –
máshā̀yā́ ‘drinkers’ (Jaggar 2001, p. 108)

As can be witnessed in (22), agentive nouns of monosyllabic and bisyllabic bases essentially assign an LH-pattern. Under a spreading account, this pattern directly generalises to trisyllabic bases, as witnessed by (21), yielding L^{+}-H.

The second piece of data illustrating spreading up to a prespecified prefixal tone is contributed by certain ethnonyms. As shown in (23), singular ethnonyms of this type[29] are marked by a short low prefix *bà-* and a tone-integrating suffix *-ē* with an HL melody for masculine, and *-iyā* with HLH for feminine.

(23) a. Fàránsà 'France' –
Bàfáránshḕ (M), Bàfáránshìyā́ (F), Fáránsā́wā́ (PL)
'French' (Newman 2000, p. 171)

b. Jā́mùs 'Germany' –
Bàjā́múshḕ (M), Bàjā́múshìyā́ (F), Jā́músā́wā́ (PL)
'German' (Newman 2000, p. 171)

(24) Dàmágàrám 'Damagaram' –
Bàdámágárḕ (M), Bàdámágárìyā́ (F), Dámágárā́wā́ (PL)
'person from Damagaram' (Newman 2000, p. 171)

With quadrisyllabic instances of this derivational pattern, we observe an LHHL surface melody: although the medial H-H sequence in (23) may already be indicative of spreading, the pentasyllabic example in (24) shows more convincingly that we are again facing spreading to the left, up to the prespecified prefixal tone.

The phenomenon of spreading up to some prespecified tone presents a challenge to the implementation of spreading proposed so far: since constraints in the formalism are inviolable,[30] the *h*-list* constraint accounting for the sequence of H will inevitably impose a *high* tone specification for the prefix, leading to a unification failure. Crysmann (2009) suggested circumventing this issue by assigning non-spreading tones in these few cases. Even if not particularly elegant, a solution along these lines is nevertheless feasible, since the complexity of the tone lists is quite limited: in fact, pentasyllabic cases are quite rare – the example in (24) appears to be the only one cited by Newman (2000) – and the existing patterns could be broken down into three sub-patterns, according to syllable count. Similar observations regarding the complexity of tone patterns can be made for

[29] Besides the pattern discussed here, which is the more common, though still not fully productive one, there is also a non-integrating tone pattern, i.e. one where base tone is preserved. See Newman (2000, p. 142) for details.

[30] The LKB does provide default constraints, which could be put to use here. However, none of the efficient processing platforms, such as PET (Callmeier 2000) or Packard's ACE (Crysmann and Packard 2012), support this.

agentive nouns. Furthermore, the formation of ethnonyms is limited in productivity, with new forms favouring an analytical construction instead:

(25) a. Pàlàsɗī́nù 'Palestine' –
ɗán/mùtúmìn Pàlàsɗī́nù (M.SG), Pálásɗī́nā́wā́ (PL)
'Palestinian' (Newman 2000, p. 174)

b. Bósníyà 'Bosnia' –
ɗán/mùtúmìn Bósníyà (M.SG), Bósníyā́wā́ (PL)
'Bosnian' (Newman 2000, p. 174)

While these observations regarding productivity and complexity are certainly valid, they can hardly obscure the fact that we would miss a generalisation here. Even if it can be shown, with respect to Hausa, that this will not lead to any serious problems regarding weak generative capacity, the current solution will certainly not scale up to other tone languages that feature both spreading and prespecified prefixal and suffixal tone.

I therefore propose a generalised approach to spreading that caters for the possibility of prespecified tone. To this end, I shall integrate into the feature structure of tone lists an additional stack of tones to be assigned from the left inwards: the feature LEFT.

As depicted in Figure 16, an H tone spreading list with a leftmost L (*l-h*-list*) does not directly specify the quality of the tone on its list elements (HD), but will rather assign the elements specified on LEFT one-by-one backwards from the end of the (primary) tone list.

Figure 16: H tone spreading with prespecified initial L

$$\begin{bmatrix} \textit{l-h*-ne-list} & \\ \textsc{left} & \boxed{r} \\ \textsc{hd} & \boxed{t} \\ \textsc{tl} & \begin{bmatrix} \textit{l-h*-list} & \\ \textsc{left} & \begin{bmatrix} \textsc{hd} & \boxed{t} \\ \textsc{tl} & \boxed{r} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

(a) Recursion

$$\begin{bmatrix} \textit{l-h*-e-list} & \\ \textsc{left} & \begin{bmatrix} \textsc{hd} & \textit{low} \\ \textsc{tl} & \textit{h*-list} \end{bmatrix} \end{bmatrix}$$

(b) Termination

The prespecified tones themselves are associated with the end of the tone list: i.e. the empty list type *l-h*-e-list* specifies on the LEFT list,

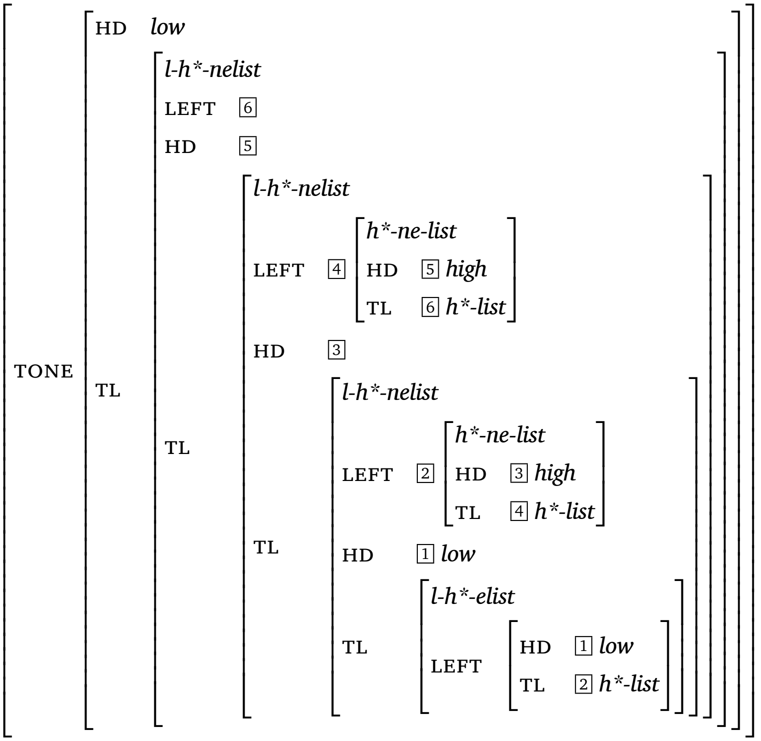


Figure 17: Sample analysis of L-H^{+}-L assignment to quadrisyllabic words

from left to right, the prefixal tones, followed by a list constraint pertaining to the spreading tone. As dictated by the constraint on the non-empty *l-h*-ne-list*, these tones are assigned one-by-one to HD, starting from the end of the tone list. Once the prefixal non-spreading tones on the LEFT list are exhausted, this constraint will insert instances of the spreading tone.

Let us briefly consider a sample analysis of a quadrisyllabic ethnonym with an L-H^{+}-L melody, as shown in Figure 17: the morphological rule of ethnonym formation will assign a final L tone to the beginning of the tone list (HD), together with an *l-h*-list* constraint on the remaining tones (TL). Owing to synchronisation with the LEN list, the tone list will be constrained to be exactly 4 elements long, thus automatically triggering recursive expansion of the *l-h*-list* constraint, specialising intermediate types to *l-h*-ne-list*, and the end of the list to *l-h*-e-list*, thereby instantiating the LEFT list with <*low* | *h*-list*>. Based on the constraints associated with the type *l-h*-ne-list*, these tones are inserted backwards, one-by-one on to the TONE list, triggering expansion of *h*-list* to *h*-ne-list* as necessary.

The generalised approach to spreading I have just proposed essentially synthesises the proposals by Newman (2000) and Leben (1978): while primary organisation is indeed right-to-left, including recursive propagation of tone-list constraints, actual instantiation with concrete surface tone constraints proceeds left-to-right. Thus, rather than stipulating a universal direction of assignment, the formal means offered by a lean unification formalism inherently constrain assignment to an outside-in regime.

5.5 *The expressive power of tone-list constraints*

We have seen so far how a purely unification-based approach to tone assignment and spreading by means of tone-list constraints can provide for an efficient implementation of morphological tone within the context of a computational grammar of Hausa. In this section, we shall investigate now on a more abstract level what functionalities the current approach provides, in general, and try to assess to what degree this approach may scale up to the computational treatment of morphological tone in general.

Owing to the pure feature structure encoding of lists, non-agglutinative operations on tone and length, i.e. deletion or modification, are generally easier to specify at the beginning of the list, rather than at its end, thereby favouring right-to-left encoding for predominantly suffixal, and left-to-right encoding for predominantly prefixal languages. Furthermore, we have established that prefixation of syllable length specifications in a suffixing language can be integrated with an overall right-to-left organisation, by means of difference lists. In the absence of spreading, the difference list approach could be straightforwardly applied to tone as well. If, however, as we have seen in Hausa, such prefixal tones are prepended to spreading tones, it is vital to feed these prefixal tone specifications as floating tones of a generalised tone spreading constraint.

Having now established that suprasegmental representations can be augmented at either end, one aspect in need of further elaboration is the question of “feature-changing” operations: while alignment with a preference regarding morphological composition will ensure that the majority of such operations will be supported by the chosen direction of suprasegmental encoding, the question still remains as to what degree such operations will be possible at the disfavoured end. Essentially there are two answers to this question: strictly regarding the treatment of prefixation formulated above, where contributions on the PREFIX difference lists are *immediately* concatenated with the suprasegmental lists of the base, subsequent feature-changing operations targeting the prefix will be inexpressible. However, a simple change regarding the principle governing list composition will greatly extend the expressive power of the current approach: instead of immediately concatenating prefixal tones, we can delay concatenation up to the next prefixal morphological rule. Once we do this, any immediately subsequent prefixation rule will have a privileged data structure to operate on, namely the suprasegmental specification of the previously attached prefix, which can then be modified or deleted, if appropriate.

To summarise, typed list constraints provide a powerful mechanism to incorporate analyses inspired by autosegmental phonology into HPSG-style computational grammars, building solely on unification. The constraints and representations I have proposed in this paper permit automatic spreading, as well as addition, modification and subtraction of tonal material at both the right and left end of the tonal representation. Crucially, this formalisation differs from traditional autosegmental approaches in two respects: first, descriptions are surface-oriented. Thus, instead of pre-linking, delinking, and re-spreading, I use direct specification of tones and melodies, plus local modification by morphological rules. Second, the association convention used here is a simple bijective relation between tones and tone-bearing units. As a consequence, there is no double association, but contour tones are represented as such. Generalisation over tone classes, e.g., *fall* and *low*, can instead be achieved by means of a tonal type hierarchy.

There is, however, one remaining limitation: owing to the monotonic nature of the constraint formalism, there can only ever be one spreading tone within a morphological domain. In Hausa, plurals of augmentative adjectives provide a case where more than one indisputable instance of spreading exists within a morphological complex.

(26) a. tánkwálḗlḕ (SG.M) – tánkwálā́-tànkwàlā̀ (PL)
'large and round' (Newman 2000, p. 25)

b. búllúƙī́ (SG.M) – búllúƙā́-bùllùƙā̀ (PL)
'huge' (Newman 2000, p. 25)

c. tsālḗlḕ (SG.M) – tsālā́-tsā̀lā̀ (PL)
'tall and slender' (Newman 2000, p. 76)

As illustrated in (26), these plurals are formed using total reduplication, assigning an H^+ tonal pattern to the base on the left and an L^+ pattern to the reduplicant on the right.

However, a purely phonological approach to reduplication is already computationally expensive, independently of the question of tone assignment: in contrast to partial reduplication, there is no principled upper bound on the number of cross-serial dependencies between segments. Thus, from a purely segmental perspective, it is advisable to approach such formations in terms of a compound structure involving two like bases. If, however, total reduplication is best viewed as involving compounding of two minimal morphological words, the instance of multiple spreading within the larger morphological complex can be broken down into two independent spreading processes within the two minimal morphological words that contribute to the compound structure. It remains to be seen whether there are languages (other than Hausa) that feature unmistakable instances of multiple spreading, i.e. extended sequences of like tones that are demonstrably independent of syllable count, outside the domain of total reduplication, or, more generally, outside morphologically compound structures.

6 CONCLUSION

In this paper, I have argued for a highly efficient encoding of suprasegmental information for the treatment of tone and length in a computational grammar of Hausa using typed list constraints. Investigating a range of morphological processes in the language, I have shown that manipulations of length and segmental material are of a highly local nature, as opposed to tonal operations that require non-local specification of melodies, including automatic spreading, in addition to agglutinative processes. I have argued that typed list constraints provide an efficient way of underspecifying spreading, and I have shown how “feature-changing” processes, such as modification and deletion, as well as prespecified tones, can be integrated with monotonic list constraints, thus covering the full range of phenomena pertaining to lexical and grammatical tone in Hausa.

REFERENCES

Mahamane L. ABDOULAYE (1992), *Aspects of Hausa Morphosyntax in Role and Reference Grammar*, Ph.D. thesis, SUNY Buffalo, NY.

Peter ADOLPHS, Stephan OEPEN, Ulrich CALLMEIER, Berthold CRYSMANN, Dan FLICKINGER, and Bernd KIEFER (2008), Some Fine Points of Hybrid Natural Language Parsing, in Nicoletta CALZOLARI, Khalid CHOUKRI, Bente MAEGAARD, Joseph MARIANI, Jan ODIJK, Stelios PIPERIDIS, and Daniel TAPIAS, editors, *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, European Language Resources Association (ELRA), Marrakech, Morocco, ISBN 2-9517408-4-0, `http://www.lrec-conf.org/proceedings/lrec2008/`.

Steven BIRD and Ewan KLEIN (1994), Phonological Analysis in Typed Feature Systems, *Computational Linguistics*, 20:455–491.

Jonathan CALDER (1989), Paradigmatic Morphology, in *Fourth Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 58–65, Association for Computational Linguistics, Manchester.

Ulrich CALLMEIER (2000), PET – A Platform for Experimentation with Efficient HPSG Processing Techniques, *Journal of Natural Language Engineering*, 6(1):99–108.

Bernard CARON (1991), *Le haoussa de l'Ader*, Dietrich Reimer, Berlin.

Bernard CARON and A. H. AMFANI (1997), *Dictionnaire français-haoussa: suivi d'un index haoussa-français*, IFRA-Ibadan, Ibadan.

Bob CARPENTER (1992), *The Logic of Typed Feature Structures with Applications to Unification-based Grammars, Logic Programming and Constraint Resolution*, volume 32 of *Cambridge Tracts in Theoretical Computer Science*, Cambridge University Press, New York.

Noam CHOMSKY and Morris HALLE (1968), *The Sound Pattern of English*, Harper and Row.

George Nicholas CLEMENTS (1985), The Geometry of Phonological Features, *Phonology Yearbook*, 2:223–252.

George Nicholas CLEMENTS and Kevin C. FORD (1979), Kikuyu Tone Shift and its Synchronic Consequences, *Linguistic Inquiry*, 10:95–108.

William F. CLOCKSIN and Christopher S. MELLISH (1981), *Programming in Prolog*, Springer, Heidelberg, 5th edition.

Ann COPESTAKE (2002), *Implementing Typed Feature Structure Grammars*, CSLI Publications, Stanford.

J. Ronayne COWAN and Russell SCHUH (1976), *Spoken Hausa*, Spoken Language Services, Ithaca.

Berthold CRYSMANN (2005a), An Inflectional Approach to Hausa Final Vowel Shortening, in Geert BOOIJ and Jaap VAN MARLE, editors, *Yearbook of Morphology 2004*, pp. 73–112, Kluwer.

Berthold CRYSMANN (2005b), Syncretism in German: a Unified Approach to Underspecification, Indeterminacy, and likeness of Case, in Stefan MÜLLER, editor, *Proceedings of the 12th Int'l Conference on Head-driven Phrase Structure Grammar (HPSG), Aug 22–24, Lisbon*, pp. 91–107, CSLI publications, Stanford.

Berthold CRYSMANN (2009), Autosegmental Representations in an HPSG for Hausa, in *Proceedings of the ACL-IJCNLP workshop on Grammar Engineering Across Frameworks (GEAF 2009)*, pp. 28–36, ACL.

Berthold CRYSMANN (2011), A Unified Account of Hausa Genitive Constructions, in Philippe DE GROOTE, Markus EGG, and Laura KALLMEYER, editors, *Formal Grammar. 14th International Conference, FG 2009, Bordeaux, France, July 25-26, 2009, Revised Selected Papers*, volume 5591 of *Lecture Notes in Artificial Intelligence*, pp. 102–117, Springer.

Berthold CRYSMANN (2012a), HaG — an Implemented Grammar of Hausa, in Michael R. MARLO, Nikki B. ADAMS, Christopher R. GREEN, Michelle MORRISON, and Tristan M. PURVIS, editors, *Selected Proceedings of the 42nd Annual Conference on African Linguistics (ACAL 42)*, pp. 321–337, Cascadilla Press, Somerville, MA.

Berthold CRYSMANN (2012b), On the Categorial Status of Hausa Genitive Prepositions, in Bruce CONNELL and Nicholas ROLLE, editors, *Selected Proceedings of the 41st Annual Conference on African Linguistics (ACAL 41)*, pp. 29–39, Cascadilla Press, Somerville, MA.

Berthold CRYSMANN (2012c), Resumption and Island-hood in Hausa, in Philippe DE GROOTE and Mark-Jan NEDERHOF, editors, *Formal Grammar. 15th and 16th International Conference on Formal Grammar, FG 2010 Copenhagen, Denmark, August 2010, FG 2011 Lubljana, Slovenia, August 2011*, volume 7395 of *Lecture Notes in Computer Science*, pp. 50–65, Springer.

Berthold CRYSMANN (2015), Resumption and Extraction in an Implemented HPSG of Hausa, in *Proceedings of the ACL-IJNLP workshop on Grammar Engineering Across Frameworks (GEAF-2015), Beijing, China*, ACL.

Berthold CRYSMANN and Woodley PACKARD (2012), Towards Efficient HPSG Generation for German, a Non-configurational Language, in *Proceedings of the*

24th International Conference on Computational Linguistics (COLING 2012), pp. 695–710, Mumbai, India.

Daniel P. FLICKINGER (2000), On Building a More Efficient Grammar by Exploiting Types, *Natural Language Engineering*, 6(1):15–28.

John A. GOLDSMITH (1976), *Autosegmental Phonology*, Ph.D. thesis, MIT.

Sharon INKELAS and William R. LEBEN (1990), Where Phonology and Phonetics Intersect: The Case of Hausa Intonation, in Mary E. BECKMAN and John KINGSTON, editors, *Between the Grammar and the Physics of Speech*, Papers in Laboratory Phonology, pp. 17–34, Cambridge University Press, New York.

Philip JAGGAR (2001), *Hausa*, John Benjamins, Amsterdam.

Herrmann JUNGRAITHMAYR and Wilhelm J. G. MÖHLIG (1976), *Einführung in die Hausa-Sprache: Kursus für Kolleg und Studium*, Reimer, Berlin.

Jean-Pierre KOENIG (1999), *Lexical Relations*, CSLI publications, Stanford.

Hans-Ulrich KRIEGER (1996), *TDL — A Type Description Language for Constraint-Based Grammars*, volume 2 of *Saarbrücken Dissertations in Computational Linguistics and Language Technology*, DFKI GmbH, Saarbrücken.

Alex LASCARIDES, Ted BRISCOE, Nicholas ASHER, and Ann COPESTAKE (1996), Order Independent and Persistent Typed Default Unification, *Linguistics and Philosophy*, 19(1):1–90.

Alex LASCARIDES and Ann COPESTAKE (1999), Default Representation in Constraint-Based Frameworks, *Computational Linguistics*, 25:55–105.

William R. LEBEN (1971), The Morphophonemics of Tone in Hausa, in C.-W. KIM and Herbert STAHLKE, editors, *Papers in African Linguistics*, pp. 201–218, Linguistic Research, Edmonton.

William R. LEBEN (1973), *Suprasegmental Phonology*, Ph.D. thesis, MIT.

William R. LEBEN (1978), The Representation of Tone, in Victoria FROMKIN, editor, *Tone: A Linguistic Survey*, pp. 177–219, Academic Press, New York.

Walt Detmar MEURERS (2001), On Expressing Lexical Generalizations in HPSG, *Nordic Journal of Linguistics*, 24(2):161–217.

Paul NEWMAN (1986), Tone and Affixation in Hausa, *Studies in African Linguistics*, 17(3):249–267.

Paul NEWMAN (2000), *The Hausa Language. An Encyclopedic Reference Grammar*, Yale University Press, New Haven, CT.

Paul NEWMAN and Philip JAGGAR (1989), Low Tone Raising in Hausa: A Critical Assessment, *Studies in African Linguistics*, 28(3):227–252.

Paul NEWMAN and Roxana MA NEWMAN (1977), *Modern Hausa–English Dictionary*, University Press, Ibadan and Zaria, Nigeria.

Stephan OEPEN, E. CALLAHAN, Daniel FLICKINGER, Christopher MANNING, and Kristina TOUTANOVA (2002), LinGO Redwoods: A Rich and Dynamic Treebank for HPSG, in *Beyond PARSEVAL. Workshop at the Third International Conference on Language Resources and Evaluation, LREC 2002*, Las Palmas, Spain.

Fred W. PARSONS (1960), The Verbal System in Hausa, *Afrika und Übersee*, 44:1–36.

Gerald PENN (2004), Balancing Clarity and Efficiency in Typed Feature Logic Through Delaying, in *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pp. 239–246, Barcelona, Spain, `http://www.aclweb.org/anthology/P04-1031`.

Carl POLLARD and Ivan SAG (1987), *Information-Based Syntax and Semantics*, volume 1, CSLI, Stanford.

Carl POLLARD and Ivan SAG (1994), *Head-Driven Phrase Structure Grammar*, CSLI and University of Chicago Press, Stanford.

Susanne RIEHEMANN (1998), Type-Based Derivational Morphology, *Journal of Comparative Germanic Linguistics*, 2:49–77.

Russell SCHUH (1989), The Reality of 'Hausa Low Tone Raising': A Response to Newman & Jaggar, *Studies in African Linguistics*, 20:253–262.

James M. SCOBBIE (1993), Constraint Violation and Conflict from the Perspective of Declarative Phonology, *Canadian Journal of Linguistics*, 38:155–167.

Ekkehard WOLFF (1993), *Referenzgrammatik des Hausa*, LIT, Münster.

Henkin semantics for reasoning with natural language

Michael Hahn[1] *and Frank Richter*[2]
[1] Eberhard Karls Universität Tübingen, Tübingen, Germany
[2] Goethe Universität Frankfurt, Frankfurt a.M., Germany

ABSTRACT

The frequency of intensional and non-first-order definable operators in natural languages constitutes a challenge for automated reasoning with the kind of logical translations that are deemed adequate by formal semanticists. Whereas linguists employ expressive higher-order logics in their theories of meaning, the most successful logical reasoning strategies with natural language to date rely on sophisticated first-order theorem provers and model builders. In order to bridge the fundamental mathematical gap between linguistic theory and computational practice, we present a general translation from a higher-order logic frequently employed in the linguistics literature, two-sorted Type Theory, to first-order logic under Henkin semantics. We investigate alternative formulations of the translation, discuss their properties, and evaluate the availability of linguistically relevant inferences with standard theorem provers in a test suite of inference problems stated in English. The results of the experiment indicate that translation from higher-order logic to first-order logic under Henkin semantics is a promising strategy for automated reasoning with natural languages.

Keywords:
Henkin semantics, reasoning, reducing higher-order reasoning to first-order reasoning

1 INTRODUCTION

One of the big challenges for applying automated inference to natural language input comes from a stark discrepancy in the preferred logical languages in theoretical semantics on the one hand and in computational semantics on the other. Theoretically-minded linguists customarily employ expressive higher-order logics in their theories of meaning in order to elegantly account for important and intricate features of the human language such as intensionality and generalized quantifiers. In contrast to these established linguistic theories, the most successful logical reasoning strategies with natural language rely on theorem provers and model builders for first-order logic. Advanced and sophisticated theories of meaning thus seem entirely out of reach for applications of automated reasoning, and any hope for adequate logic-based reasoning with language may seem doomed even before we start to consider additional challenges such as the necessary integration of world knowledge and discourse pragmatics.

To cope with the discrepancy, previous work by Bos and Markert (2006) on applying first-order inference tools to natural language in part approximated intensions and higher-order quantification in first-order logic, and in part ignored their role in language. Of course, this strategy restricts the fragment of natural language that can be treated, and it forces computational semanticists to recast the theories of formal linguists in a different logical language. Analyses of intensional contexts in terms of possible worlds can be simulated in first-order logic by adding worlds to the first-order structure (Lewis 1968), but some generalized quantifiers such as ‘most’, when given a plausible formal definition of their meaning, can be shown not to be expressible in first-order logic (Barwise and Cooper 1981).

However, on second glance, all hope is not lost for wielding the higher-order descriptions of formal semanticists in computational environments in a more direct, systematic and comprehensive fashion. A standard approach to automated inference with higher-order logic outside of linguistics exploits a reduction to first-order logic that is complete for *Henkin semantics*, a semantics for higher-order logic that is weaker than the standard semantics, but for which complete proof systems exist. In this paper, we explore the application of this idea to natural language input, starting our inferencing toolchain with logical representations for natural language sentences couched in two-sorted Type Theory (Ty2, Gallin 1975), one of the standard higher-order logics favored by formal semanticists.

The inferencing architecture we will introduce thus avoids an error-prone and ad-hoc case-by-case approximation of higher-order phenomena that requires a separate verification of the adequacy of each hand-encoded solution. Instead reasoning starts with the original higher-order representations of linguists, which are reduced to first-order logic by a systematic translation with well-understood properties. Rather than being tailored to specific linguistic applications, the present proposal provides a general translation of full higher-order logic, and the fine-grained semantic representations of the formal semantics literature are accepted as input without any modification. This means that higher-order representations of challenging natural language facts can be developed independently of implementations in formalisms familiar to semanticists without having to worry about a possible manual reduction to first-order logic, and the original representations may then serve as input to automated reasoning. Since the semantics of higher-order logic is preserved in the translation process (in a sense to be elucidated shortly), the first-order translation is guaranteed to be adequate for any input.

We begin by defining a Henkin semantics for Ty2 and illustrating how it differs from its standard semantics, arguing that Henkin semantics is not only formally interesting but also adequate for reasoning with natural language. After defining two translations of different logical strength from higher-order logic to first-order logic and describing their mathematical properties, we assess the practical value of the general strategy outlined above with a test suite of natural-language inference problems that focuses on phenomena that have figured prominently in linguistics. Test items that encode reasoning problems in natural language are translated into Ty2 under standard semantic analyses derived from Montague's seminal PTQ fragment of English (Montague 1973), and from there into first-order logic by our Henkin-complete translation function. We then apply standard first-order inference tools to evaluate the feasibility of automated reasoning on the resulting first-order translations.

In Section 2, we introduce the formal definition of Henkin semantics and argue that it provides much of the proof-theoretic strength needed to formalize linguistically relevant natural language inferences. Section 3 defines a systematic translation from Ty2 to first-order logic and its properties. Section 4 is devoted to the evaluation of the approach, presenting a grammar fragment with meaning postulates, the test suite, and the results of our experiments. In the remaining sections we discuss related work and future perspectives. The appendix contains axioms for the translations, essential proofs of their properties, meaning postulates for lexical items in our grammar, and the test suite.

2 HENKIN SEMANTICS FOR TY2

Validity in first-order logic is semi-decidable, while validity in higher-order logic is not. It follows that there can be no computable translation from higher-order logic to first-order logic that is both sound and complete for standard semantics. However, Henkin (1950) showed that higher-order logic can be given a natural semantics such that the valid formulae are exactly the theorems of a certain formal calculus. As all semi-decidable problems are reducible to first-order theorem proving, the task of proving higher-order theorems for *Henkin semantics* can in principle be reduced to first-order theorem proving.

While not every higher-order tautology is valid for Henkin semantics, we will demonstrate that certain linguistically interesting higher-order theorems that are not expressible in first-order logic indeed are. This observation subsumes, for instance, many natural inferences licensed by the quantifier 'most', which is undefinable in first-order logic.

2.1 *Ty2*

We assume two-sorted Type Theory (Ty2), a standard language for formalizing semantic analyses for natural language (see, e.g., Groenendijk and Stokhof 1982), as representation language.

Classical type theory as formulated by Church (1940) has only two basic types, *e* for entities and *t* for truth values (ι and o in Church's notation). Ty2 has two basic types apart from *t*, namely *e* for entities and *s* for possible worlds. Since classical type theory consists of those expressions of Ty2 in which types containing *s* do not occur, our translation below can also be applied to analyses in classical one-sorted higher-order logic.

Let us first define the syntax of Ty2, and Henkin semantics. The presentation essentially follows Gallin (1975).

Definition 1. *Types is the smallest set such that:*

- $s, e, t \in$ *Types,*
- *if* $\sigma, \tau \in$ *Types, then* $\langle \sigma\tau \rangle \in$ *Types.*

t is the type of the truth values *true* and *false*, s is the type of possible worlds, and e the type of entities. $\langle\sigma\tau\rangle$ is the type of functions mapping objects of type σ to objects of type τ. We let c^n be an enumeration of the words over some finite alphabet.

Definition 2 (Syntax of Ty2). *The set $\mathscr{L}_{\mathrm{Ty2}}$ of Ty2 terms is the smallest set such that:*

- *for every type τ and every $n \in \mathbb{N}$, $x^n_\tau \in \mathscr{L}_{\mathrm{Ty2}}$ (variables),*
- *for every type τ and every $n \in \mathbb{N}$, $c^n_\tau \in \mathscr{L}_{\mathrm{Ty2}}$ (constants),*
- *if $\alpha_{\langle\sigma\tau\rangle}$ and β_σ are in $\mathscr{L}_{\mathrm{Ty2}}$, then $(\alpha\beta)_\tau$ is in $\mathscr{L}_{\mathrm{Ty2}}$ (function application),*
- *if α_τ is in $\mathscr{L}_{\mathrm{Ty2}}$, then $(\lambda x^n_\sigma \alpha_\tau)_{\langle\sigma\tau\rangle}$ is in $\mathscr{L}_{\mathrm{Ty2}}$ for every $n \in \mathbb{N}$ (lambda abstraction).*

For every type σ, the constants $\dot{\forall}^\sigma_{\langle\langle\sigma t\rangle t\rangle}$ (universal quantifier over objects of type σ), $\dot{\exists}^\sigma_{\langle\langle\sigma t\rangle t\rangle}$ (existential quantifier), $\iota^\sigma_{\langle\langle\sigma t\rangle\sigma\rangle}$ (choice operator), and $\dot{=}^\sigma_{\langle\sigma\langle\sigma t\rangle\rangle}$ (equality) are constants of $\mathscr{L}_{\mathrm{Ty2}}$.[1] Moreover, $\dot{\neg}_{\langle tt\rangle}$, $\dot{\wedge}_{\langle t\langle tt\rangle\rangle}$, $\dot{\vee}_{\langle t\langle tt\rangle\rangle}$ and $\dot{\rightarrow}_{\langle t\langle tt\rangle\rangle}$ are constants of $\mathscr{L}_{\mathrm{Ty2}}$. The dots are intended to prevent confusion with the corresponding logical symbols of first-order logic. Furthermore, for all types σ, τ, ρ, we assume the combinator symbols $\mathbf{I}^\sigma_{\langle\sigma\sigma\rangle}$, $\mathbf{K}^{\sigma,\tau}_{\langle\sigma\langle\tau\sigma\rangle\rangle}$, and $\mathbf{S}^{\rho,\sigma,\tau}_{\langle\langle\rho\langle\sigma\tau\rangle\rangle\langle\langle\rho\sigma\rangle\langle\rho\tau\rangle\rangle\rangle}$. These are all *logical constants*. In addition, there is a countably infinite supply of *non-logical constants* for every type.

As the definition indicates, we write $\alpha, \beta, \ldots$ for meta-variables for terms, c for meta-variables for constants, and σ, τ for meta-variables for types. Terms of the form $(\lambda x\alpha)$ are called lambda abstracts.

We will also need weaker versions of Ty2 that contain fewer terms:

Definition 3 (Restrictions of Ty2). *Let $\mathscr{C}$ be a non-empty set of variables, constants and lambda abstracts. The language $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$, the restriction of Ty2 to $\mathscr{C}$, is the smallest set such that:*

- *$\mathscr{C} \subseteq \mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$,*
- *whenever $\alpha \in \mathscr{C}$, then every sub-term of α is also in $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$,*
- *if $\alpha_{\langle\sigma\tau\rangle}$ and β_σ are in $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$, then $(\alpha\beta)_\tau$ is in $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$ (function application).*

[1] The superscript types are regarded part of the constants' names. Generally, we formalize polymorphic constants as families of constants whose names contain the type parameters.

Following standard practice in formal semantics, we employ some abbreviations and conventions that make Ty2 terms look more similar to first-order formulae: Types are omitted where redundant. $\forall x_\sigma$ and $\exists x_\sigma$ denote $\dot{\forall}^\sigma \lambda x_\sigma$ and $\dot{\exists}^\sigma \lambda x_\sigma$, respectively. Variables are often represented by letters other than x, and without a number superscript. In particular, variables of type s are often denoted by w, and variables over propositions, properties and other higher-order objects by P or Q. Functional application is written in an 'uncurried' functional notation: $P(x, y(z))$ stands for $((Px)(yz))$. Logical constants such as $\dot{\rightarrow}$ and $\dot{\equiv}$ are rendered in infix notation.

2.2 Henkin semantics

Henkin semantics was originally introduced by Henkin (1950) for one-sorted type theory, but the generalization to Ty2 is straightforward (Gallin 1975, p. 59). We will use a more general variant in which the universes for higher types may be empty, which will be needed for a weak version of our translation.

Definition 4. *A frame D is a collection of mutually disjoint sets $\{D_\sigma\}_{\sigma \in Types}$ such that:*

1. *D_e, $D_s \neq \emptyset$,*
2. *$D_t \subseteq \{T, F\}$,*
3. *for $\langle \sigma \tau \rangle \in$ Types, $D_{\langle \sigma \tau \rangle}$ is a (possibly empty) set of functions from D_σ to D_τ.*

An $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-assignment v with respect to a frame D is a mapping from $\mathscr{V}^{\mathscr{C}}$, the variables of $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$, into the domain of D such that variables of type σ are mapped to elements of D_σ.[2] *An $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-interpretation $\mathscr{I}$ is a mapping from the constants of $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$ to D such that constants of type σ are mapped to elements of D_σ, and, for every type σ, the following conditions hold:*

1. *If $\dot{\forall}^\sigma \in \mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$, then $\mathscr{I}(\dot{\forall}^\sigma)(\mathsf{x}) = T$ if and only if $\mathsf{x}(\mathsf{y}) = T$ for every $\mathsf{y} \in D_\sigma$.*
2. *If $\dot{\equiv}^\sigma \in \mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$, then $\mathscr{I}(\dot{\equiv}^\sigma)(\mathsf{x})(\mathsf{y}) = T$ if and only if $\mathsf{x} = \mathsf{y}$ ($\mathsf{x}, \mathsf{y} \in D_\sigma$).*
3. *If $\dot{\neg} \in \mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$, then $\mathscr{I}(\dot{\neg})(\mathsf{x}) = T$ if $\mathsf{x} = F$ and F if $\mathsf{x} = T$.*
4. *If $\dot{\wedge} \in \mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$, then $\mathscr{I}(\dot{\wedge})(\mathsf{x})(\mathsf{y}) = T$ if and only if $\mathsf{x} = \mathsf{y} = T$.*

[2] Note that $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-assignments w.r.t. D only exist when $D_\tau \neq \emptyset$ for all $x^n_\tau \in \mathscr{C}$; *mutatis mutandis*, the same holds for interpretations.

5. *Analogous definitions are assumed for other connectives and* $\dot{\exists}^\sigma$.
6. *If* $\iota^\sigma \in \mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$, *and* $\mathrm{g} \in D_{\langle\sigma t\rangle}$ *is such that* $\mathrm{g}(\mathrm{x}) = T$ *for some* $\mathrm{x} \in D_\sigma$, *then* $\mathrm{g}(\mathscr{I}(\iota^\sigma)(\mathrm{g})) = T$. *Otherwise,* $\mathrm{g}(\mathscr{I}(\iota^\sigma)(\mathrm{g})) = F$.
   Informally, ι *selects an element out of every non-empty set. Because of this property,* ι *is called a choice operator.*
7. *Appropriate equations are assumed for the combinators.*[3]

Given a frame D, *an* $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$*-interpretation* $\mathscr{I}$, *and an assignment* v, *the interpretation function* $[\![\cdot]\!]^v_{D,\mathscr{I}}$ *is defined on the terms of* $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$ *by induction as follows:*

1. *if* $x^n_\sigma \in \mathscr{V}^{\mathscr{C}}$, *then* $[\![x^n_\sigma]\!]^v_{D,\mathscr{I}} := v(x^n_\sigma)$,
2. *if* $c^n_\sigma \in \mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$ *is a constant, then* $[\![c^n_\sigma]\!]^v_{D,\mathscr{I}} := \mathscr{I}(c^n_\sigma)$,
3. *if* $(\alpha_{\langle\sigma\tau\rangle}\beta_\sigma)_\tau \in \mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$, *then* $[\![\alpha\beta]\!]^v_{D,\mathscr{I}} := [\![\alpha]\!]^v_{D,\mathscr{I}}\left([\![\beta]\!]^v_{D,\mathscr{I}}\right)$,
4. *if* $(\lambda x^n_\sigma \alpha_\tau)_{\sigma\tau} \in \mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$, *then* $[\![(\lambda x^n_\sigma \alpha_\tau)_{\sigma\tau}]\!]^v_{D,\mathscr{I}} :=$ *the function* $f : D_\sigma \to D_\tau$ *such that* $f(\mathrm{x}) := [\![\alpha_\tau]\!]^{v[x^n_\sigma \mapsto \mathrm{x}]}_{D,\mathscr{I}}$,[4]

where the third and the fourth clause result in an undefined value if $[\![\beta]\!]^v_{D,\mathscr{I}} \notin D_\sigma$, *or if* $[\![\alpha]\!]^v_{D,\mathscr{I}}$ *is not defined.*

Definition 5. *A frame* D *with an* $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$*-interpretation* $\mathscr{I}$ *is called a general* $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$*-model if, for every* $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$*-assignment* v *and every term* $\alpha_\sigma \in \mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$, $[\![\alpha_\sigma]\!]^v_{D,\mathscr{I}}$ *is a well-defined element of* D_σ. *A term* α_t *is called a* $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$*-Henkin tautology iff* $[\![\alpha_t]\!]^v_{D,\mathscr{I}} = T$ *for all general* $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$*-models* D *and all* $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$*-assignments* v.

If $\mathscr{C}$ *contains all logical constants, we refer to* $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$*-Henkin tautologies simply as Henkin tautologies, and to general* $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$*-models as general* $\mathscr{L}_{\mathrm{Ty2}}$*-models.*

A general $\mathscr{L}_{\mathrm{Ty2}}$*-model* $\langle D, \mathscr{I}\rangle$ *is called a full model if, for every* $\langle\sigma\tau\rangle \in$ *Types,* $D_{\langle\sigma\tau\rangle}$ *contains all functions from* D_σ *to* D_τ. *A term* α_t *is called a tautology in the standard sense if* $[\![\alpha_t]\!]^v_{D,\mathscr{I}} = T$ *for all full models* $\langle D, \mathscr{I}\rangle$ *and all assignments* v.

[3] When $\mathscr{C}$ does not contain all lambda abstracts, the presence of the (finitely many) combinators still yields the full strength of Henkin semantics as it is usually defined. However, as we will mostly be concerned with weaker versions of Henkin semantics, the combinators play no role for our immediate purposes, and we omit the equations for reasons of space. See Hindley and Seldin 2008, p. 110, for the necessary equations.

[4] Note that $v[x^n_\sigma \mapsto \mathrm{x}]$ is a $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-*assignment*, as $x^n_\sigma \in \mathscr{C}$.

Informally, there are two types of models for higher-order logic. *Full models* are defined by the requirement that they contain any higher-order function over their domain that in principle exists. *General $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-models* are only required to provide *some* interpretation for every term of $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$. The notion of 'general model' is in turn graded by the set $\mathscr{C}$: having more elements in $\mathscr{C}$ results in a stronger semantics, i.e., a semantics that allows fewer models.

Every full model is also a general $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-model for every $\mathscr{C}$, but the converse does not hold: some general $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-models are not full models. Similarly, if $\mathscr{C} \subseteq \mathscr{D}$, then every general $\mathscr{L}^{\mathscr{D}}_{\mathrm{Ty2}}$-model is also a general $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-model. There is an inverse relationship between the sets of tautologies for the various notions of semantics. Since every full model is a general model, all $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-Henkin tautologies hold in every full model and are therefore tautologies in the standard sense. However, there are tautologies in the standard sense that are not $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-Henkin tautologies for any $\mathscr{C}$. Analogously, if $\mathscr{C} \subseteq \mathscr{D}$, then all $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-tautologies are also $\mathscr{L}^{\mathscr{D}}_{\mathrm{Ty2}}$-tautologies. It is in this sense that the notion of 'general model' yields a weaker semantics than the standard semantics of higher-order logic, and that increasing $\mathscr{C}$ results in a stronger semantics. A semantics based on full models is called a *standard semantics*, and a semantics based on general models is a *Henkin semantics*. When $\mathscr{C}$ contains all logical constants, our definitions of 'general models' and 'Henkin tautologies' coincide with the usual definition of Henkin semantics, since all lambda abstracts can be defined with the combinators (Hindley and Seldin 2008, p. 110). The significance of Henkin semantics for our application rests on the following theorem of Henkin (1950):

Theorem 6 (Henkin's Completeness Theorem). *There is a (finitary) calculus that generates exactly the set of Henkin tautologies of* $\mathscr{L}_{\mathrm{Ty2}}$.

Because the set of tautologies in the standard sense is not recursively enumerable, no such theorem is available in the standard case.

Example

Let us assume that $\mathscr{C} = \{woman_{\langle et\rangle}, dance_{\langle et\rangle}\} \cup \{x^n_\tau : n \in \mathbb{N}, \tau \in \{e, t, \langle ee\rangle, \langle et\rangle, \langle tt\rangle, \langle t\langle tt\rangle\rangle, \langle\langle et\rangle t\rangle, \langle\langle et\rangle\langle\langle et\rangle t\rangle\rangle\}\}$. Consider the frame characterized by the following sets:

- $D_e := \{a, b, c, d, e\}$,
- $D_{\langle ee\rangle} := \{\{a \mapsto a, b \mapsto a, c \mapsto a, d \mapsto a, e \mapsto a\}\}$,

- $D_{\langle et\rangle}, D_{\langle tt\rangle}, D_{\langle t\langle tt\rangle\rangle}, D_{\langle\langle et\rangle t\rangle}, D_{\langle\langle et\rangle\langle\langle et\rangle t\rangle\rangle}$ are the full sets of functions with the respective domains and ranges,
- for other types τ, we set $D_\tau := \emptyset$.

For this frame, we define an interpretation $\mathscr{I}$ given by $\mathscr{I}(woman) = \chi_{D_e}$ and $\mathscr{I}(dance) = \chi_{\{a,b\}}$. Since D_τ is empty for most τ, the frame does not constitute a full model. However, the frame together with $\mathscr{I}$ is a general $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-model.

We now add a constant $most_{\langle\langle et\rangle\langle\langle et\rangle t\rangle\rangle}$, representing the natural-language quantifier 'most'. 'Most' is often assumed to express that more than half of the elements in the restrictor are also in the nuclear scope (e.g., Gamut 1991, p. 252, Westerståhl 2011). In $most(woman, dance)$, $woman$ is the restrictor and $dance$ is the nuclear scope, and the term is true iff more than half of the women are also dancers. Barwise and Cooper (1981, C13) show that, under this interpretation, the meaning of 'most' cannot be expressed in first-order logic.[5] However, it is definable in $\mathscr{L}_{Ty2}$. Generalized to sets of any cardinality, MOST(P,Q) is true if and only if the cardinality of $P \cap Q$ is strictly greater than that of $P \backslash Q$. Equivalently, MOST(P,Q) is true if and only if $P \cap Q \neq \emptyset$ and there is no surjective mapping from $P \backslash Q$ to $P \cap Q$. If we identify subsets of D_e with their characteristic functions, i.e., the functions of type $\langle et\rangle$, we can express this definition in $\mathscr{L}_{Ty2}$ as follows:

(1) $\forall P_{\langle et\rangle} \forall Q_{\langle et\rangle} \ : \ most(P,Q) \leftrightarrow [\exists x_e(P(x) \wedge Q(x)) \wedge$
$\forall f_{\langle ee\rangle} : (\forall y_e : (P(y) \wedge \neg Q(y)) \rightarrow (P(f(y)) \wedge Q(f(y))))$
$\rightarrow \exists x_e(P(x) \wedge Q(x) \wedge \forall z_e : (P(z) \wedge \neg Q(z)) \rightarrow f(z) \neq x)]$
'MOST(P,Q) holds if and only if $P \cap Q \neq \emptyset$, and
for every mapping f from $P \backslash Q$ to $P \cap Q$
there is an $x \in P \cap Q$ that is not in the image of $P \backslash Q$ under f'
(i.e., f is not surjective)

[5] The intuition is that, when describing the cardinality of a set using a formula of first-order logic, one can only count up to some fixed finite number which depends on the formula, not being able to distinguish sets of greater cardinality. Choosing for each first-order formula (1) a sufficiently large universe of a model M and (2) sets U and V such that $U, V, M \backslash U, M \backslash V$, and the relative difference of U and V are each sufficiently large, we see that MOST(U, V) cannot be first-order definable. For the version generalized to infinite sets that we will consider, the undefinability follows more easily from the compactness theorem.

In the sense of the standard semantics of higher-order logic, this definition indeed ensures that MOST has the desired model-theoretic interpretation: in a full model that satisfies (1), $\mathscr{I}(most)(\chi_P, \chi_Q)$ holds if and only if the cardinality of $P \cap Q$ is strictly greater than that of $P \backslash Q$.

This is not always true for general models. To see this, consider the general $\mathscr{L}^{\mathscr{C}}_{\text{Ty2}}$-model we constructed. Since $D_{\langle\langle et\rangle\langle\langle et\rangle t\rangle\rangle}$ contains all possible functions, it includes in particular a function such that (1) becomes true when $\mathscr{I}(most)$ is set to this function. Under this interpretation, the term $((most\ woman)\ dance)$ is true in the model: the single function $f : D_e \to D_e$ included in the frame maps $\mathscr{I}(woman) \backslash \mathscr{I}(dance)$ to $\{a\} \subsetneq \mathscr{I}(woman) \cap \mathscr{I}(dance)$. In other words, there is no surjective function in this particular general model from $P \backslash Q$ to $P \cap Q$, as required for *most* according to (1). However, intuitively the statement 'most women dance' is not satisfied in the model: $\mathscr{I}(woman)$ contains five elements, while $\mathscr{I}(dance)$ only contains two elements. Thus, even when the definition of 'most' is satisfied in a general model, it need not actually have the intended model-theoretic interpretation. The problem with definitions like (1) is that, in a general model, the domain of the quantifier $\forall^{\langle ee\rangle}$ is not the set of functions from D_e to D_e. Instead it is $D_{\langle ee\rangle}$, which need not contain all functions from D_e to D_e.

Henkin semantics comes closer to standard semantics when $\mathscr{C}$ contains more terms – in particular, if $\mathscr{C}$ contains all logical constants of $\mathscr{L}_{\text{Ty2}}$, then every general $\mathscr{L}^{\mathscr{C}}_{\text{Ty2}}$-model for which D_e and D_s are finite is a full model. The reason is that every function between two finite sets D_σ and D_τ is definable with the choice operator. On the other hand, if D_e or D_s is infinite, then by the Löwenheim-Skolem theorem there will always be general $\mathscr{L}^{\mathscr{C}}_{\text{Ty2}}$-models which are not full models, and in which the interpretation of 'most' differs from the one intended.

2.3 *Henkin semantics for natural language*

The fact that MOST cannot be defined in a model-theoretically adequate way in Henkin semantics might be taken as evidence that it is too weak to express the concept 'most' meaningfully. But this is not the case. Many interesting facts about MOST are logical consequences of (1) under Henkin semantics. A case in point is monotonicity, one of the properties of generalized quantifiers that have received significant attention in linguistics (Barwise and Cooper 1981; Westerståhl 2011). MOST is upward monotonic in the second argument:

Proposition 7. *MOST is upward monotonic in the second argument:*
In $\mathscr{L}_{\mathrm{Ty2}}$: $\forall P_{et} Q_{et} B_{et} : (most(P,Q) \wedge (\forall x_e : Q(x) \rightarrow B(x))) \rightarrow most(P,B)$
Informally: If MOST(P,Q) *and* $B \supseteq Q$ *hold, then* MOST(P,B) *holds.*

The upward monotonicity of MOST in its second argument corresponds to the linguistic observation that the inferences in (2) are valid. In fact, different quantifiers exhibit different inference patterns, showing that these monotonicity properties are both interesting and non-trivial. For instance, 'few' does not license the parallel pattern (3):

(2) a. Most children are playing in the street. $\vdash$ Most children are playing.
b. Most men sing and dance. $\vdash$ Most men dance.

(3) a. Few children are playing in the street. $\nvdash$ Few children are playing.
b. Few men sing and dance. $\nvdash$ Few men dance.

A system for automated reasoning from natural language should account for these facts. Proposition 7 (and, by extension, formalizations of the inferences in (2)) are consequences of (1) under Henkin semantics. To see this, consider the following elementary argument:

Proof. Let $\langle D, \mathscr{I} \rangle$ be a general $\mathscr{L}_{Ty2}$-model in which MOST(P,Q) and $B \supseteq Q$ hold. Clearly, $P \cap B \supseteq P \cap Q \neq \emptyset$. Let $f \in D_{\langle ee \rangle}$ with $\{f(x) : x \in P \backslash B\} \subseteq P \cap B$. For all $x \in D_e$, set $\pi(x)$ to be x if $x \in P \cap Q$ and an arbitrary element of $P \cap Q$ otherwise. Define $f' : D_e \rightarrow P \cap Q$ by $f'(x) := \pi(f(x))$. As a suitable π can be defined in $\mathscr{L}_{\mathrm{Ty2}}$ with the choice operator ι,[6] $f' \in D_{\langle ee \rangle}$. By the assumption MOST$(P,Q)$, we know that $f'|_{P \backslash Q} : P \backslash Q \rightarrow P \cap Q$ is not surjective. Thus, $f|_{P \backslash B} : P \backslash B \rightarrow P \cap B$ cannot be surjective. As f was arbitrary, MOST(P,B) holds. □

As we only assumed that $\langle D, \mathscr{I} \rangle$ is a general $\mathscr{L}_{\mathrm{Ty2}}$-model, Proposition 7 is a Henkin tautology. It should also be noted that the proof crucially relies on the choice operator, and the proposition does indeed not hold in all general $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-models if $\mathscr{C}$ is too small.

[6] Informally, we use lambda abstraction to define $A_x := \{x\} \cap P \cap Q$ if $x \in P \cap Q$ and $A_x := P \cap Q$ otherwise. Then we set $\pi(x) := \iota(A_x)$. Formally, $\pi := [\![\lambda x^1_e . \iota^e \lambda x^2_e \left(x^3(x^2) \wedge x^4(x^2) \wedge ((x^3(x^1) \wedge x^4(x^1)) \rightarrow x^1 = x^2) \right)]\!]^{\nu[x^3 \mapsto \chi_P,\ x^4 \mapsto \chi_Q]}_{D,\mathscr{I}}$.

A similar argument shows that Henkin semantics is also strong enough to prove that, if there are at least four objects of type e, MOST is not downward monotonic in either argument, and not upward monotonic in the first argument. It is also possible to formalize more specific numerical inferences, such as 'If exactly four out of five members of P are also in Q, then most members of P are in Q.' Three of the four semantic postulates for 'most' given by Barwise and Cooper (1981, p. 208)[7] are provable from our definition of 'most' under Henkin semantics as well.

It may seem surprising that Henkin semantics is strong enough to prove non-trivial facts about MOST, even though it cannot define it in a model-theoretically adequate way. The point is that many consequences of (1) do not depend on the existence of functions that are not definable by lambda abstraction, and are, for that reason, true in every general $\mathscr{L}_{\text{Ty2}}$-model.[8] This is an instance of a general phenomenon. Although concrete mathematical theorems can be constructed that are true in all full models but not valid for Henkin semantics, we are not aware of any known theorem of this kind that is not of a meta-mathematical nature and is interesting to mathematicians working outside of logic. Given this it seems plausible that Henkin semantics provides all the proof-theoretical strength that is needed for typical natural language inferences.

3 TRANSLATING TY2 TO FIRST-ORDER LOGIC

The crucial step for leveraging the power of first-order reasoning engines when coming from semantic representations in higher-order logic is in the formulation of an appropriate translation from higher-

[7] In our notation, they are the following: (1) MOST(A, A) always holds, (2) upward monotonicity in the second argument, (3) if $A \neq \emptyset$, then MOST(A, X) is true for some but not all sets X, (4) if $A \neq \emptyset$, then MOST(A, X) and MOST(A, Y) together imply $X \cap Y \neq \emptyset$. (4) follows under the standard semantics, the other three postulates also follow under Henkin semantics. To be precise, our definition proves MOST(A, A) only under the assumption $A \neq \emptyset$, as MOST$(\emptyset, \emptyset)$ is false according to our axiom. Depending on whether one views it as being intuitively true or false, our axiom could be modified to evaluate MOST$(\emptyset, \emptyset)$ to true.

[8] As opposed to consequences that do depend on the existence of such functions and, for that reason, are only guaranteed to hold in full models; they may be false in some general models.

order logic to first-order logic. In this section, we will show how the translation from Ty2 to first-order logic can be effected in such a way that Henkin tautologies are translated into first-order tautologies. The guiding idea is that the translation of terms of Ty2 into terms and formulae of first-order logic preserves the term structure as faithfully as possible and aims at exploiting the strengths of first-order provers by translating terms representing connectives and quantifiers into the corresponding symbols of first-order logic. Moreover, two groups of first-order axioms are added in the process that encode the typing and the intended behavior of the translations of Ty2 terms. Given these axioms the translations of Henkin tautologies are theorems of first-order logic – the translation is *complete* for Henkin semantics. It is also *sound*: if the translation of a term is a first-order tautology, then the term itself must be a Henkin tautology.

3.1 *Translation*

The translation consists of three parts: a type translation T_{ty} (translating types into first-order terms), a term translation T_{term} (translating terms of Ty2 into terms of first-order logic), and a formula translation T_f (translating Ty2 terms of type t into first-order formulae). The overall translation T of a term of type t is obtained as its formula translation with the addition of two groups of axioms. The components of the translation will be described next.

Types are represented by first-order terms. The basic types t, s, e are directly represented by first-order constants. Higher types are represented by terms of first-order logic by replacing $\langle\cdot\cdot\rangle$ by $g(\cdot,\cdot)$ as follows:

(4) a. $T_{ty}(\tau) := \tau$ if $\tau = e, s, t$

b. $T_{ty}(\langle\sigma\tau\rangle) := g(T_{ty}(\sigma), T_{ty}(\tau))$

The idea behind the translation of Ty2 terms is that terms of type t are translated into first-order formulae, and other terms into first-order terms. We first define a *term translation* T_{term}, translating every Ty2 term into a first-order term. Polymorphic constants are represented by first-order functions whose arguments represent their type arguments. For instance, ι^e is translated as $\iota(e)$, where ι is a one-place first-order function symbol. Other constants and variables are translated as themselves: $T_{term}(c_\tau) := c$, where c is a first-order constant, and $T_{term}(x_\tau) := x$. Note that while variables and constants are translated as themselves, the type information attached to the terms is not directly accessible to the first-order language and will be encoded in additional axioms.

Functional application is translated recursively as a two-place function symbol:

(5) $T_{term}(\alpha\beta) := f(T_{term}(\alpha), T_{term}(\beta))$

Lambda abstracts are translated by introducing a function symbol whose arguments represent the free variables of the lambda abstract. Formally, assume that we are given a term $\lambda x.\alpha$ with free variables $\{(v_1)_{\sigma^1}, ..., (v_n)_{\sigma^n}\}$. Then $T_{term}(\lambda x_\tau.\alpha_\sigma) := g_{\lambda x.\alpha}(v_1, ..., v_n)$, where $g_{\lambda x.\alpha}$ is a fresh n-place function symbol which by itself does not carry any meaning. Its intended behavior will be encoded in an additional axiom.

Our third translation function, *formula translation* (T_f), is only applied to terms of type t; it translates them into first-order formulae. Propositional connectives, quantifiers, and the equality operator are translated into the corresponding logical symbols of first-order logic whenever possible. The remaining terms of type t are converted to first-order formulae using the predicate symbol *isTrue*:

(6) a. $T_f((\dot{\circ}\alpha_t)\beta_t) := T_f(\alpha) \circ T_f(\beta)$
if $\circ$ is a binary propositional connective

b. $T_f(\dot{\neg}\alpha_t) := \neg T_f(\alpha)$

c. $T_f(\dot{\forall}^\tau(\lambda x_\tau.\alpha_t)) := \forall x : \mathit{hasType}(x, T_{ty}(\tau)) \rightarrow T_f(\alpha)$ (similarly for $\exists$)

d. $T_f(\dot{\forall}^\tau \alpha_{\langle \tau t\rangle}) := \forall x : \mathit{hasType}(x, T_{ty}(\tau)) \rightarrow \mathit{isTrue}(f(T_{term}(\alpha), x))$
if α is not a lambda abstract (similarly for $\exists$)

e. $T_f((\dot{\equiv}^t\alpha)\beta) := (T_f(\alpha) \leftrightarrow T_f(\beta))$

f. $T_f((\dot{\equiv}^\tau\alpha)\beta) := (T_{term}(\alpha) = T_{term}(\beta))$ if $\tau \neq t$

g. $T_f(\alpha_t) := \mathit{isTrue}(T_{term}(\alpha))$ when no other case applies

If α_t is a term of type t, the overall translation $T(\alpha)$ is defined to be the formula translation $T_f(\alpha)$.

To illustrate the definitions, let us consider the term in (7a). This term straightforwardly encodes an (extensional) translation of the sentence ‘Most men sing and dance.’ The actual Ty2 term behind the simplified notation in (7a) is (7b). By the preceding definitions, its term translation is (7c). Thus, the overall (formula) translation, as given by (6g), is the first-order formula (7d).

(7) a. $most(man, \lambda x_e.sing(x) \wedge dance(x))$

b. $((most\ man)\ \lambda x_e.\phi)$ with $\phi = (\dot{\wedge}\ (sing\ x))\ (dance\ x)$

c. $T_{term}((most\ man)\ \lambda x_e.\phi)$
$= f(f(most(e), man), T_{term}(\lambda x_e.\phi))$ (by 5)
$= f(f(most(e), man), g_{\lambda x_e.\phi})$

d. $isTrue(f(f(most(e), man), g_{\lambda x_e.\phi})$

Note that the term translation of $most_{\langle\langle et\rangle\langle\langle et\rangle t\rangle\rangle}$ as $most(e)$ follows from assuming that *most* is a polymorphic constant $most^{\sigma}_{\langle\langle\sigma t\rangle\langle\langle\sigma t\rangle t\rangle\rangle}$ with type argument $\sigma = e$ in our example.

Axioms

To ensure that translations of Henkin tautologies are in fact provable, axioms need to be added that encode the meaning and the intended behavior of the function and predicate symbols. They are stated in the first-order language of the translation.

Type information is not encoded in the first-order translation of Ty2 terms. A first group of axioms guarantees the correct typing of all objects. For instance the following axiom states that the result of applying a functor of type $\langle\sigma\tau\rangle$ to an argument of type σ has type τ:

(8) $\forall x_0 \forall x_1 \forall x_2 : [\exists x_3(hasType(x_0, g(x_3, x_2))\ \wedge\ hasType(x_1, x_3))]$
$\rightarrow hasType(f(x_0, x_1), x_2)$

A second group of axioms encodes postulates of Henkin’s system, defining the types and the intended behavior of constants and functions. For instance, given a type τ, the next axiom states that the translation $\iota(T_{ty}(\tau))$ of the iota operator ι^{τ} selects an element from every non-empty set of objects of type τ:

(9) $\forall y : hasType(y, g(T_{ty}(\tau), t)) \rightarrow$
$\big[(\exists\ z\ isTrue(f(y, z))) \rightarrow isTrue(f(y, f(\iota(T_{ty}(\tau)), y)))\big]$
‘For every object y of type $\langle\tau, t\rangle$ such that $y(z) = T$ for some z, $y(\iota^{\tau}(y)) = T$ also holds.’

For every function symbol $g_{\lambda x.\alpha}$, there is an axiom which states that, given arguments of the appropriate types, the function has the value defined by the lambda abstract. More formally, assuming that the free variables of α are $\{v^1_{\sigma_1}, ..., v^n_{\sigma_n}\}$, the axiom takes the form (10a) if α is of type t, and (10b) otherwise:

(10) a. $\forall v_1, ..., v_n : (hasType(v_1, T_{ty}(\sigma^1)) \land \ ... \ \land hasType(v_n, T_{ty}(\sigma^n)))$
$\rightarrow [hasType(g_{\lambda x.\alpha}(v_1, ..., v_n), g(T_{ty}(\tau), t))$
$\land \forall x : hasType(x, T_{ty}(\tau))$
$\rightarrow [isTrue(f(g_{\lambda x.\alpha}(v_1, ..., v_n), x)) \longleftrightarrow T_f(\alpha)]]$

b. $\forall v_1, ...v_n$: $[hasType(v_1, T_{ty}(\sigma^1)) \land ... \land hasType(v_n, T_{ty}(\sigma^n))] \rightarrow$
$[hasType(g_{\lambda x.\alpha}(v_1, ..., v_n), g(T_{ty}(\tau), T_{ty}(\sigma)))$
$\land \forall x$: $hasType(x, T_{ty}(\tau)) \rightarrow f(g_{\lambda x.\alpha}(v_1, ..., v_n), x) = T_{term}(\alpha)]$

In the case of example (7c), the defining axiom of $g_{\lambda x_e.\phi}$ is (11).

(11) $hasType(g_{\lambda x_e.\phi}, g(e, t)) \land \forall x : hasType(x, e) \rightarrow$
$\big[isTrue(f(g_{\lambda x_e.\phi}, x)) \longleftrightarrow (isTrue(f(sing, x)) \land isTrue(f(dance, x)))\big]$

Given $\mathscr{C} \subset \mathscr{L}_{\mathrm{Ty2}}$, we define the $\mathscr{C}$-axiomatization, $\mathscr{A}^{\mathscr{C}}$, as the set containing the first group of axioms (for typing) and the defining axioms for all constants, variables, and lambda abstracts in $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$. The intention is that $\mathscr{A}^{\mathscr{C}}$ provides the necessary information to prove $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-Henkin tautologies.[9] We can show that this is indeed the case:[10]

Theorem 8. *Let $\mathscr{C} \subset \mathscr{L}_{\mathrm{Ty2}}$ and $\alpha_t \in \mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$. Then α is an $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-Henkin tautology if and only if $\mathscr{A}^{\mathscr{C}} \vdash T(\alpha)$.*

In this sense our translation is sound and complete for Henkin semantics.

3.2 *Restricting the axiomatization*

The strength of Henkin semantics and, in consequence, the usefulness of the first-order translation of higher-order meaning characterizations of natural language expressions depends on the choice of $\mathscr{C}$. On the one hand, we have seen that choosing $\mathscr{C}$ to be too

[9] The full set of axioms can be found in Appendix A.

[10] A proof is given in Appendix B.1.

small may result in linguistically relevant inferences not being covered. On the other hand, when automated reasoning techniques come into play, a surplus of axioms may easily distract the algorithms, making automated inference inefficient to the point of being practically infeasible.

In our experiments to be described in the next section we will use two axiomatizations. For a term α_t, the *strong axiomatization* $\mathscr{A}^s(\alpha)$ is constructed from the set $\mathscr{C}$ containing all constants, variables, and lambda abstracts in α and, furthermore, all logical constants of Ty2. If $\mathscr{C}$ contains instances of a polymorphic constant, such as ι^τ for some type τ, a single axiom is used for all types, like (12), replacing the infinitely many axioms in (9). This choice keeps $\mathscr{A}^s(\alpha)$ finite. Due to the combinators, the strong axiomatization has the full strength of Henkin semantics.

(12) $\forall x \forall y : \mathit{hasType}(y, g(x,t)) \rightarrow$
$[(\exists z\ \mathit{isTrue}(f(y,z))) \rightarrow \mathit{isTrue}(f(y, f(\iota(x), y)))]$

The *weak axiomatization* $\mathscr{A}^w(\alpha)$ is constructed from the set $\mathscr{C}(\alpha)$ that contains only the lambda abstracts, variables, and constants occurring in α. If α contains instances of a polymorphic constant, only axioms for those specific instances occurring in α are used. We can go even further and leave out constants and lambda abstracts when they are eliminated by the formula translation. More precisely, we add logical constants to $\mathscr{C}(\alpha)$ only when they are translated into corresponding first-order constants rather than into first-order connectives or quantifiers. Similarly, lambda abstracts enter $\mathscr{C}(\alpha)$ only when they do not exclusively occur as arguments of constants which represent first-order quantifiers.

Unlike the strong axiomatization, the weak axiomatization lacks the full power of Henkin semantics, but it also has considerable advantages. As it introduces fewer axioms than the translation with strong axiomatization, it might remove an unnecessary burden from the theorem provers. Where they fail for the strong axiomatization, they might still be able to prove theorems of weak translations of semantic representations of natural language. More importantly, under certain conditions the weak translation has finite models, which is highly relevant and desirable in the context of automated reasoning, since finite models can be constructed automatically.[11] We obtain the following theorem:[12]

Theorem 9. *Assume $\alpha_t \in \mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$ is true in a general $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-model for which D_e and D_s are finite. Then $\bigwedge_{\phi \in \mathscr{A}^w(\alpha)} \phi \wedge T(\alpha)$ is true in some finite (first-order) model.*

However, note that satisfiability of the weak translation of α_t only ensures satisfiability in a $\mathscr{L}^{\mathscr{C}(\alpha)}_{\mathrm{Ty2}}$-model, not necessarily satisfiability in a $\mathscr{L}_{\mathrm{Ty2}}$-model. In the context of our application, this means that weak translations of Henkin validities might not be provable, and that models for weak translations may not correspond to $\mathscr{L}_{\mathrm{Ty2}}$-models.

3.3 *Relationship to previous translations*

The translation we outlined in this section is similar to previous translations from higher-order logic to first-order logic, in particular to the 'lambda lifting' translation of Meng and Paulson (2008) and to the translation of Hurd (2002), who also encodes types as first-order terms. Compared to approaches which represent types by means of first-order predicate symbols (Kerber 1992), typing by terms offers the advantage that it can be expressed with finitely many axioms. This is of course crucial for the application of automated reasoning tools. The main difference between our formulation and Meng and Paulson (2008) resides in the special treatment of connectives and quantifiers in the formula translation, T_f. Treating the logical constants separately makes it possible to exploit the strengths of first-order theorem provers at the inferencing step. Unlike Hurd (2002) and Meng and Paulson (2008), we provide a formal proof of soundness and completeness (Appendix B.1).

4 TESTING AND EVALUATION

We have defined a translation from Ty2 to first-order logic and made precise in which sense it preserves the semantics of Ty2. To assess the feasibility of automated inference on the resulting first-order formulae in a linguistic context, we now apply our translation and standard

[11] The possibility of finite models may be surprising at first, as even $\mathscr{A}^w(\alpha)$ seems to model an infinite set of types. However, note that $\mathscr{A}^w(\alpha)$ does not contain an axiom that demands that e, s, and t and the higher types be distinct.

[12] A proof is given in Appendix B.2.

first-order reasoning engines to a set of natural language reasoning problems of the type commonly considered in linguistics. Our selection focuses on sentences with lexical elements whose semantic analysis involves intensionality and generalized quantifiers, because these are typically cited as the main motivation for higher-order logic rather than first-order logic in the semantic characterization of natural language expressions. A sizable part of our test suite is derived from the FraCaS test suite (Cooper *et al.* 1996), which was created precisely for evaluating the semantic competence of natural language processing systems.

We will first introduce a fragment of English with Montague-style semantic representations and standard meaning postulates for (classes of) lexical elements from the literature. We then describe the contents and structure of our test suite and proceed to assess the performance of first-order inference engines (comprising theorem provers and model builders) on the task of classifying valid and invalid inferences that are handed to them in the form of first-order translations of the higher-order logical representations which our grammar assigns to the test items. The inference engines of course also draw on the meaning postulates as additional axioms. Throughout our evaluation, we will disregard complications arising from possible ambiguities and only consider pre-determined intended readings of our items.

The experiments will show that the weak translation performs significantly better than the strong translation, confirming or refuting 87.7% of those items in the test suite where a proof or refutation exists in principle. While every item that is challenging due to intensionality is correctly recognized, items involving generalized quantifiers are considerably harder for automated reasoning.

4.1 *Fragment*

Our fragment is derived from the English textbook grammar of Blackburn and Bos (2005), who construct semantic representations directly in first-order logic. Their grammar architecture is well-suited for our purposes because its modular design easily supports alternative semantic representation languages by simply plugging in other lexical semantic specifications and adding syntactic rules where needed. Moreover, Blackburn and Bos' grammar is already equipped with an interface to different reasoning engines that we can exploit for evaluating the performance of inference engines on our first-order translations.

The semantic analyses are inspired by Montague's PTQ fragment (Montague 1973), with two major changes: As laid out in the previous sections, we use Ty2 rather than Intensional Logic (IL). Ty2 offers technical advantages (Friedman and Warren 1980, p. 323), and, as a version of typed lambda calculus, its formal properties are well-understood. Montague's representations can be translated straightforwardly into Ty2, since IL can be regarded a sublanguage of Ty2 (Gallin 1975). Second, we follow Bennett (1974) and Dowty *et al.* (1981, p. 188) in representing the arguments of extensional predicates as individuals rather than individual concepts. For instance, *walk* is translated into a term of type $\langle s\langle et\rangle\rangle$, while Montague chose the more elaborate $\langle s\langle\langle se\rangle t\rangle\rangle$.

Representative lexical entries are shown in Figure 1. They are mostly standard. 'Believe' and 'know' take as their arguments a possible world, a proposition, and an entity that represents the agent (Montague 1973). Adverbs attaching to VPs map properties to properties. We translate the definite article by means of the ι operator, i.e., a choice function (von Heusinger 1997). We opt for a uniform treatment of all adjectives as functions mapping properties to properties, following Montague (1970). Generalized quantifiers are rendered as functions of type $\langle\langle et\rangle\langle\langle et\rangle t\rangle\rangle$, i.e., as relations between sets of objects of type e. The fragment licenses both singular and plural noun phrases, but no special plural semantics is assumed; the occurrence of plurals is restricted to NPs with quantifiers such as 'most' and 'many'.

The context-free grammar rules of the fragment stipulate how the semantic representations of daughter constituents are combined to derive the semantic representation of their mother node. The typical mode of composition is functional application. The phrase structure rules of our grammar needed for the test suite are shown in Figure 2 together with their semantic composition rules. The fragment generates one translation per syntactic analysis and does not account for scope ambiguities. This is not a substantial restriction since ambiguities could be captured by adopting one of Blackburn and Bos' alternatives of semantic composition with a more sophisticated underspecified semantics by means of dominance constraints. Our choice here is

Cat.	Words	Translation
V_{intr}	dance	$dance_{\langle s\langle et\rangle\rangle}$
V_{tr}	see	$\lambda P_{\langle\langle s\langle et\rangle\rangle t\rangle}\lambda w_s\lambda x_e.P(\lambda w_s\lambda y_e\ see_{\langle s\langle e\langle et\rangle\rangle\rangle}(w_s,x_e,y_e))$
$\mathrm{V}_{i\text{-}tr}$	seek	$\lambda P_{\langle s\langle\langle s\langle et\rangle\rangle t\rangle\rangle}\lambda w_s\lambda x_e.seek_{\langle s\langle\langle s\langle\langle s\langle et\rangle\rangle t\rangle\rangle\langle et\rangle\rangle\rangle}(w,P,x)$
V_{cop}	be	$\lambda P_{\langle\langle s\langle et\rangle\rangle t\rangle}\lambda w_s\lambda x_e.P(\lambda w_s\lambda y_e\ (x=y))$
V_{s}	know	$\lambda P_{\langle st\rangle}\lambda w_s\lambda x_e.know_{\langle s\langle\langle st\rangle\langle et\rangle\rangle\rangle}(w,P,x)$
V_{aux}	does not	$\lambda P_{\langle s\langle et\rangle\rangle}\lambda w_s\lambda x_e.\neg P(w,x)$
Adv	possibly	$\lambda P_{\langle s\langle et\rangle\rangle}\lambda w^1_s\lambda x_e.possibly_{\langle s\langle\langle st\rangle t\rangle\rangle}(w^1,\lambda w^2_s P(w^2,x))$
Adj	tall	$\lambda P_{\langle s\langle et\rangle\rangle}\lambda w_s\lambda x_e.tall_{\langle s\langle\langle s\langle et\rangle\rangle\langle et\rangle\rangle\rangle}(w,P,x)$
Det	most	$\lambda P_{\langle s\langle et\rangle\rangle}\lambda Q_{\langle s\langle et\rangle\rangle}.most_{\langle\langle et\rangle\langle\langle et\rangle t\rangle\rangle}(P(w_s),Q(w_s))$
	every	$\lambda P_{\langle s\langle et\rangle\rangle}\lambda Q_{\langle s\langle et\rangle\rangle}.\forall x_e(P(w_s,x)\rightarrow Q(w_s,x))$
	the	$\lambda P_{\langle s\langle et\rangle\rangle}\lambda Q_{\langle s\langle et\rangle\rangle}.Q(w_s,\iota^e(P(w_s)))$
PN	John	$\lambda P_{\langle s\langle et\rangle\rangle}.P(w_s,john_e)$
N	unicorn	$unicorn_{\langle s\langle et\rangle\rangle}$
P	in	$in_{\langle\langle\langle s\langle et\rangle\rangle t\rangle\langle\langle s\langle et\rangle\rangle\langle s\langle et\rangle\rangle\rangle\rangle}$
Conj	and	$\lambda P_{\langle s\langle et\rangle\rangle}\lambda Q_{\langle s\langle et\rangle\rangle}\lambda w_s\lambda x_e.(P(w,x)\wedge Q(w,x))$

Figure 1: Lexical Entries. For every category, an example word is given

S:$\alpha\beta$ → NP:α VP:β

VP:α → V_{intr}:α

VP:$\alpha\beta$ → V_{tr}:α NP:β

VP:$\alpha(\lambda w_s.\beta)$ → $\mathrm{V}_{i\text{-}tr}$:α NP:β

VP:$\alpha(\lambda w_s.\beta)$ → V_s:α S:β

VP:$\lambda x\exists P_{\langle s\langle e\rangle\rangle}\alpha(P,w,x)$ → V_{cop} Adj:α

VP:$\alpha\beta$ → V_{cop}:α NP:β

VP:$\beta(\alpha,\gamma)$ → VP:α Conj:β VP:γ

VP:$\alpha\beta$ → V_{aux}:α VP:β

VP:$\alpha\beta$ → Adv:α VP:β

VP:$\alpha\beta$ → VP:β PP:α

PP:$\alpha\beta$ → P:α NP:β

NP:$\alpha\beta$ → Det:α N:β

NP:α → PN:α

N:$\alpha\beta$ → Adj:α N:β

Figure 2: Phrase Structure Rules

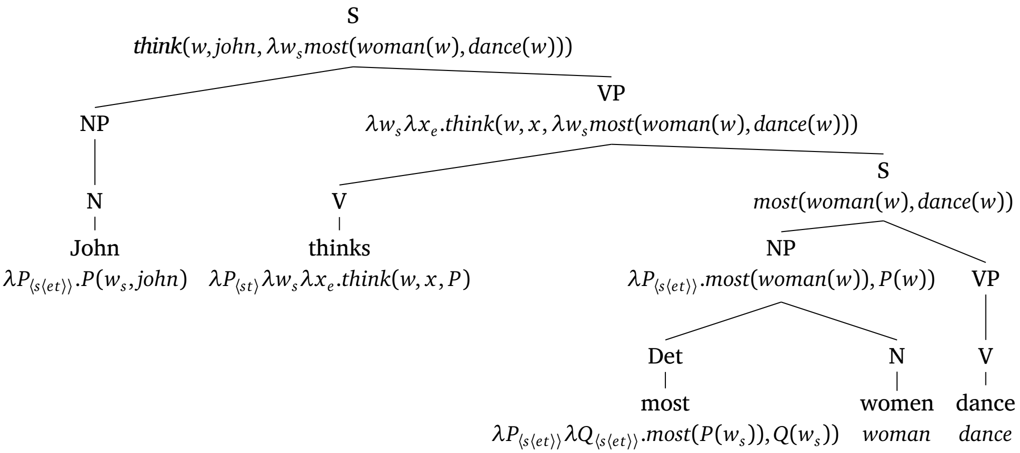

S
$think(w, john, \lambda w_s most(woman(w), dance(w)))$
NP
N
John
$\lambda P_{\langle s\langle et\rangle\rangle}.P(w_s, john)$
VP
$\lambda w_s \lambda x_e.think(w, x, \lambda w_s most(woman(w), dance(w)))$
V
thinks
$\lambda P_{\langle st\rangle} \lambda w_s \lambda x_e.think(w, x, P)$
S
$most(woman(w), dance(w))$
NP
$\lambda P_{\langle s\langle et\rangle\rangle}.most(woman(w)), P(w))$
VP
Det
most
$\lambda P_{\langle s\langle et\rangle\rangle} \lambda Q_{\langle s\langle et\rangle\rangle}.most(P(w_s)), Q(w_s))$
N
women
$woman$
V
dance
$dance$

Figure 3: An analysis in our fragment

motivated by simplicity and the compatibility of the easiest choice of composition mechanism with our main objectives.

The analysis of the sentence ‘John thinks most women dance’ is shown in Figure 3. The following translations illustrate the coverage of the semantic fragment:[13]

(13) Mia possibly dances.
$possibly(w^1_s, \lambda w^2_s.dance(w^2_s, mia_e))$

(14) Mia thinks that John dances.
$think(w^1_s, mia_e, \lambda w^2_s.dance(w^2_s, john_e))$

(15) Most men sing and dance.
$most(man(w_s), \lambda x_e(sing(w_s, x_e) \wedge dance(w_s, x_e)))$

(16) The blond man dances.
$dance(w_s, \iota^e_{\langle\langle et\rangle e\rangle}(blond(w_s, man)))$

(17) John seeks a unicorn.
$seek(w_s, john_e, \lambda w_s \lambda P_{\langle s\langle et\rangle\rangle} \exists x_e\ (unicorn(w_s, x_e) \wedge P(w_s, x_e)))$

[13] With the notation of arguments we follow the linguistic convention of putting subjects before objects to enhance readability. The relationship to a strict Ty2 representation should be transparent.

4.2 *Meaning postulates*

The semantic representations we obtain from the grammar are insufficient for drawing inferences that go beyond simple first-order tautologies expressed in natural language. A substantive portion of the semantic import of words such as ‘most’ and ‘believe’ is hidden behind inconspicuous Ty2 constants such as *most* and *believe*. Since these constants by themselves are atomic expressions with arbitrary meaning, further information about their actual meaning must be made available for exploitation in reasoning.

There are two ways to add the relevant information: either by stating meaning postulates in $\mathscr{L}_{\mathrm{Ty2}}$ and adding them as axioms, or by restricting the class of models to those where the interpretations of the constants satisfy certain restrictions. A prominent example of the first option is Montague (1973);[14] the second option was chosen in the semantic postulates of Barwise and Cooper (1981) and in the treatment of generalized quantifiers in Discourse Representation Theory (Kamp and Reyle 1993, Def. 4.24). In the present context, an axiomatic solution is to be preferred as it makes it possible to enlist our translation functions to also translate potentially higher-order meaning postulates to first-order logic. In effect, the first-order translations of the postulates may simply be added to the axiomatization $\mathscr{A}$ of the first-order translation. The situation is more complicated if the information is supplied model-theoretically, as Henkin’s completeness theorem need not remain true if the class of permissible models is constrained. Therefore, we opt for the first solution and supply information about constants such as *believe* and *most* by meaning postulates in $\mathscr{L}_{\mathrm{Ty2}}$.

To see the impact of meaning postulates on reasoning and to appreciate the relevance of the translation functions for them, we discuss a selection of postulates for representative constants in our fragment.[15] The examples will also indicate the sorts of difficult semantic questions which have to be addressed in formulating appropriate axioms.

[14] Montague actually understood meaning postulates as constraints on the interpretations, but the completeness theorem for Henkin semantics guarantees that this is equivalent to treating them as axioms.

[15] The full set of meaning postulates is given in Appendix C.

Verbs: belief and knowledge

There is a considerable amount of work on the logic of knowledge and belief from a philosophical point of view (cf. Hintikka 1962; Rescher 2005 for an overview). It has often been argued that belief and knowledge should be closed under logical inferences (Rescher 2005). We assume a principle of logical omniscience which states that if an agent knows (thinks) something, she knows everything which follows from it logically (see (18)). Such a postulate is not without problems as no actual person could be aware of every logical truth, but we accept it as a general consequence of the standard possible-worlds analysis of propositional attitudes. We also assume that only true propositions can be known (Rescher 2005), as formalized in (19):

(18) Deductivity Axiom

a. $\forall x_e \forall P_{\langle st\rangle} \forall Q_{\langle st\rangle} \forall w^1_s : \mathit{think}(w^1, P, x) \rightarrow$
$\big(\forall w^2_s (P(w^2) \rightarrow Q(w^2))\big) \rightarrow \mathit{think}(w^1, Q, x)$

b. ‘If x knows/believes P in world w^1 and $P \rightarrow Q$ holds necessarily, then x knows/believes Q in world w^1.’

(19) Veridicality Axiom

a. $\forall x_e \forall P_{\langle st\rangle} \forall w_s (\mathit{know}(w, P, x) \rightarrow P(w))$

b. ‘If x knows P in world w, then P is true in world w.’

Adjectives

Adjectives are commonly classified based on the inference patterns they license (Kamp and Partee 1995, Partee 1995). As examples like (20a) show, adjectives like ‘blond’ are *intersective* (see (20b)). This property is formalized by the meaning postulate (20c) (Partee 1995, p. 324).

(20) a. Mia is a blond woman. Mia is a robber. $\vdash$ Mia is a woman and Mia is a blond robber.

b. $[\![\mathit{blond}\ \mathrm{N}]\!] = [\![\mathit{blond}]\!] \cap [\![\mathrm{N}]\!]$

c. For each intersective adjective meaning ADJ:
$\exists P_{\langle s\langle et\rangle\rangle} \forall w_s \forall Q_{\langle s\langle et\rangle\rangle} \forall x_e : \mathrm{ADJ}(w, Q, x) \leftrightarrow [P(w, x) \wedge Q(w, x)]$
where P, which is uniquely defined by the axiom, represents the set $[\![\mathrm{blond}]\!]$ in (20b).

Similar meaning postulates account for other *subsective* and for *privative* adjectives (Partee 1995).[16]

Other adjectives, such as 'alleged', 'potential', and 'arguable', are neither subsective nor privative. They do not allow any inference on whether the property denoted by the noun holds: an 'alleged robber' may or may not be a robber. For the modal modifier 'alleged' we adopt the following postulate, adapted from Jespersen and Primiero (2013, p. 104):

(21) a. $\forall P_{\langle s\langle et\rangle\rangle} \forall x_e^1 \forall w_s^1 : \mathit{alleged}(w^1, P, x^1)$
$\leftrightarrow \exists x_e^2\, \mathit{allege}(w^1, x^2, \lambda w^2(P(w^2, x^1)))$

b. 'Somebody alleges that x is a P if and only if x is an alleged P.'

Adverbs

It is often assumed that 'necessarily' can be modeled via universal quantification over possible worlds (Montague 1973). This is formalized by the following postulate (Gamut 1991, p. 201, MP7):

(22) a. $\forall w_s^1 \forall P_{\langle st\rangle} \left(\mathit{necessarily}(w^1, P) \leftrightarrow \forall w_s^2 P(w^2)\right)$

b. 'P is necessarily true if and only if it is true in all worlds.'

'Possibly' is characterized by replacing the universal quantifier by an existential quantifier. Note that the world argument w_s^1 plays no role and is only needed because we assume a uniform analysis of all adverbs, and the extension of many adverbs does depend on the world.

Generalized quantifiers

In (1) we saw how 'most' can be defined in Ty2. Certain quantifiers have straightforward definitions in first-order logic. Besides 'all' and 'some', these include, for instance, 'exactly two', 'at most two', and 'only'. (23a) provides a (simplistic) definition of 'only' as in 'Only men danced':

(23) a. $\forall P_{\langle et\rangle} \forall Q_{\langle et\rangle} (\mathit{only}(P, Q) \leftrightarrow \forall x_e (Q(x) \rightarrow P(x)))$

b. 'ONLY(P,Q) holds if and only if $Q \subseteq P$.'

[16] Subsective adjectives are a superclass of intersective adjectives. They license the inference that the property denoted by the noun holds: a skillful writer is a writer, but need not be a skillful violinist even if she is known to be a violinist. Privative adjectives such as 'fake' license the inference that the property denoted by the noun does not hold: a fake diamond is not a diamond.

There are other quantifiers whose meaning is less straightforward to capture, including 'few' and 'many'. However, we can indirectly characterize these quantifiers by postulating rules concerning properties such as monotonicity (Barwise and Cooper 1981, p. 209). While our two examples are upward and downward monotonic, respectively, in the second argument (Barwise and Cooper 1981, p. 185, SP 2), it is less clear whether they are also monotonic in the first argument (Barwise and Cooper 1981, p. 185). We assume that they are (see (24a)). We also state that 'few' and 'many' are incompatible (see (24b)), and postulate that 'few' holds if the intersection of its two arguments is empty (see (24c)). These axioms are somewhat weaker than the optional axiom SP4 (NOT MANY ⟺ FEW) of Barwise and Cooper (1981, p. 209), which seems unnatural to us.

(24) a. i. FEW is downward-monotonic in both arguments.
ii. MANY is upward-monotonic in both arguments.
b. $\neg(\text{FEW}(P,Q) \land \text{MANY}(P,Q))$
c. $P \cap Q = \emptyset \rightarrow \text{FEW}(P,Q)$

The axioms, whose rendering in $\mathscr{L}_{Ty2}$ is similar to what we saw for 'most' in (1), suffice to prove important facts about these quantifiers and to make relevant predictions on the validity of natural language inferences. For instance, they entail that MANY is true if the extension of the second argument is 'large', while FEW is true if it is 'small', and that MANY is *conservative*, i.e., inferences like 'Many men dance' ⇒ 'Some men dance' are valid, which corresponds to Axiom SP6 in Barwise and Cooper (1981, p. 209). Conversely, they predict that 'No men dance' is incompatible with the statement 'Many men dance'.

Considering the gradual and context-dependent nature of these two quantifiers, not many more inferences seem possible without appealing to a notion of discourse context.

Example

As an illustration of the interaction of the grammar fragment, the meaning postulates, and the translations from Ty2 to first-order logic, consider (25), a variant of one of the examples in (2). The Ty2 translations generated by our fragment for the premise and the conclusion are given in (26). The respective first-order translations are shown in (27).

(25) Most women sing and dance. ⊢ Most women dance.

(26) a. $most(woman(w), \lambda x(sing(w,x) \wedge dance(w,x))$

b. $most(woman(w), dance(w))$

(27) a. $isTrue(f(f(most, f(woman, w)), g_\phi(w)))$
where g_ϕ is defined by
$\forall w : hasType(w,s) \to [hasType(g_\phi(w), g(e,t))$
$\wedge\ \forall x : (hasType(x,e) \to ([isTrue(f(f(sing,w),x))$
$\wedge\, isTrue(f(f(dance,w),x))] \longleftrightarrow isTrue(f(g_\phi(w),x))))]$

b. $isTrue(f(f(most, f(woman, w)), f(dance, w)))$

To show that the inference in (25) is valid, we need to prove that, given the axioms and the first-order translations of the meaning postulates, (27a) logically entails (27b). We need the meaning postulate for *most*, whose first-order translation is too complex to be easily readable, but whose meaning is essentially captured by the informal explanation in (1). At this point the problem of proving the entailment in (25) has been reduced to proving a first-order formula which consists of the elements of $\mathscr{A}$, the translation of (1) and (27a) as its premises, and of (27b) as its conclusion.

For the proof, one may first remodel the higher-order proof of Proposition 7 in the first-order translation. The defining axiom of g_ϕ can then be exploited to prove that $isTrue(f(g_\phi(w),x))$ entails $isTrue(f(f(dance,w),x))$, which corresponds to the fact that 'x sings and dances' logically entails 'x dances'. By the first-order version of (1), the claim (27b) follows.

4.3 *Test suite*

We created a small test suite for natural language inference which requires solving inference problems that have figured prominently in formal semantics research. Inferences relying on world knowledge typical for prominent tasks such as the Recognizing Textual Entailment challenges (Dagan *et al.* 2009) are not addressed with our test suite, because we are interested in the feasibility and quality of reasoning with first-order translations of higher-order meaning specifications of natural language rather than in the bigger (and even more intricate) question of modeling typical human reasoning by means of other types of knowledge resources. The test suite contains 117 items divided into six sections which focus on *modality*, *knowledge and belief*, *generalized quantifiers*, *adjectives*, *de dicto readings*, and *first-order inferences*, respectively.

Each item consists of a set of *premises*, a *conjecture*, and a symbol connecting those two. The items are grouped in three classes: If the premises entail the conjecture, the inference is *valid*. If the premises are incompatible with the conjecture, we call the inference *contradictory*. Items for which the correctness of the inference is not determined by their form or by meaning postulates are *contingent*; the inferences that they represent might be supported by some models but not by others.[17] In each item the conjecture is separated from the premises by a symbol that indicates the class to which the item belongs. If the inference is valid, the separator is '⊢'; '⊢ NON' designates contradictory items, and '⋉' appears in contingent items. Consider the following example:

(28) Mia is a woman. Mia dances. ⊢ A woman dances.

The first two sentences are the premises, 'A woman dances' is the conjecture. The conjecture is entailed by the premise, as indicated by the symbol '⊢'.[18]

52 items are valid (44.4%), 12 are contradictory (10.3%), and 53 are contingent (45.3%). These judgments are based on whether the Ty2 representations of the sentences that are provided by our grammar fragment support the inference under standard higher-order semantics or not, assuming the meaning postulates as axioms.[19] For every one of our items, its membership in the three inference classes coincides for standard semantics and Henkin semantics. One item, (3.24), which tests for monotonicity properties of 'most', is special in that it requires the strong axiomatization. For all others the weak axiomatization is sufficient.

[17] Logically speaking, this means that the inferences in the last class are also invalid (like those in the second class).

[18] Our *conjecture* corresponds to what the literature on Textual Entailment calls *hypothesis* (Dagan *et al.* 2009). The subtle difference in terminology is meant to stress that there is a deeper difference in the conception of what exactly constitutes reasoning with natural language.

[19] To put it differently, the inference patterns follow the linguistic theory our fragment of English implements.

29 items in the sections on adjectives and generalized quantifiers are derived from the FraCaS test suite. The original FraCaS test suite contains question-answer pairs, but it was later converted by MacCartney (2009) to the format employed in our test suite. The following example from FraCaS together with MacCartney's conversion illustrates the difference:

(29) Original FraCaS format (item 197)

a. Premise: John has a genuine diamond.

b. Question: Does John have a diamond?

c. Answer: Yes

(30) Converted: John has a genuine diamond. $\vdash$ John has a diamond.

For our experiments, we draw on those parts of FraCaS that are covered by our fragment. It captures 18% (14 out of 80 items) of the FraCaS section on generalized quantifiers and 65% (15 out of 23 items) of the section on adjectives. Other items would require nontrivial additions to syntax or lexical items for which there is no standard analysis in the semantics literature.

In three instances the predictions implied by our grammar and meaning postulates do not align with those assumed in FraCaS: in two cases additional information about the expression 'on time' would be needed to infer that finishing on time implies finishing. In the third case the reason for the deviation is due to different assumptions about the properties of certain adjectives in FraCaS compared to what our meaning postulates assume. It is important to note that these differences between the predictions of our fragment and the FraCaS annotation result from differences in linguistic modeling, not from a weakness of Henkin semantics – given our meaning postulates, the predictions are the same for Henkin semantics and standard semantics.

4.4 *Experiment*

The goal of the experiment was to assess to what extent the translation supports efficient automated inference on reasoning problems that are typically encountered in formal semantics research. To this end we applied first-order reasoners to the natural language inference problems in the test suite. The transformation pipeline from the test suite to the application of inference engines is straightforward: The inference problems encoded in the test suite were translated to Ty2 by parsing the natural language sentences in the test items according to our grammar fragment. This step resulted in a syntactic analysis coupled with higher-order logical representations. The latter served as input to the translation to first-order logic introduced in Section 3. The first-order formulae were then ready to be processed by freely available first-order reasoners, following the implementation developed by Bos (2004).

Following Bos, two types of reasoning tools were employed: *theorem provers* and *finite model builders*. The theorem provers try to find a *proof* for each first-order formula, while the model builders try to construct a finite *model*. A complete theorem prover will find a proof for every valid formula, and it will find a proof for the negation of every contradictory formula. Thus, ideally, a proof or refutation can be found for the first-order translation of every valid or contradictory inference. However, by complexity and undecidability results of first-order logic, proving even a small formula may take a very long time, and there is no general algorithm determining whether or not a formula has a proof. In particular, there is no general procedure for showing that a formula is contingent. Finite model builders provide a partial solution to this problem: If a formula is contingent, there exist models for both the formula and its negation. If they are finite, these models can be found by a model builder. Since statements made in natural language often concern situations involving finitely many objects, it may be expected that the restriction to finite models is not critical and that for many inference problems either a proof or a counter-model is found in a reasonable amount of time. If this is the case, an automated decision of many natural-language inferences is possible.

Let us now take a closer look at the technicalities involved in putting this idea to work. As indicated earlier, the premises and the conjecture of each test item were translated to Ty2 representations according to the specifications of the grammar fragment. The *premise term* of a test item is the term $p := [\alpha_1 \wedge ... \wedge \alpha_n \wedge \beta_1 \wedge ... \wedge \beta_k]$, where $\alpha_1, ..., \alpha_n$ are the meaning postulates, and $\beta_1, ..., \beta_k$ are the Ty2 translations of the premises 1 to k. Let γ be the Ty2 translation of the conjecture. Using the taxonomy introduced in the previous section, an inference is valid if and only if $p \rightarrow \gamma$ is a tautology. It is contradictory if and only if $p \rightarrow \neg\gamma$ is a tautology. If neither of these cases holds, the inference pattern is contingent.

The inference engine constructed from theorem provers and model builders was tasked to determine if $\mathscr{A} \vdash T(p \rightarrow \gamma)$ or $\mathscr{A} \vdash T(p \rightarrow \neg\gamma)$ holds, with $\mathscr{A}$ either the strong or weak axiomatization. Table 1 summarizes the questions to the inference engine and their possible answers. Obtaining an answer is of course constrained by the general undecidability of the questions in first-order logic, entailing the risk of non-terminating searches. The theorem provers try to find a proof for either $T(p \rightarrow \gamma)$ or $T(p \rightarrow \neg\gamma)$ given the axioms $\mathscr{A}$. If the inference pattern is contingent, no proof will be found. The model builder tries to find a finite model for either $\mathscr{A} \cup \{T(p \wedge \gamma)\}$ (the inference is not contradictory) or $\mathscr{A} \cup \{T(p \wedge \neg\gamma)\}$ (the inference is not valid). If translations of natural language expressions are well-behaved for our purposes, a proof or refutation is found whenever an inference is valid or contradictory, and both a model and a countermodel are found whenever an inference pattern is contingent. Under these circumstances it is possible to determine if an item is valid, contingent, or contradictory.

	Valid	Contingent	Contradictory
$\mathscr{A} \vdash T(p \rightarrow \gamma)$	proof	–	–
$\mathscr{A} \vdash T(p \rightarrow \neg\gamma)$	–	–	proof
$\mathscr{A} \cup \{T(p \wedge \gamma)\}$	model	model	–
$\mathscr{A} \cup \{T(p \wedge \neg\gamma)\}$	–	model	model

Table 1:
Maximal possible output for valid, contingent, and contradictory inference patterns

With our experiments we are interested in determining how well our first-order translation of typical natural language reasoning problems behaves in supporting these decisions with currently available standard first-order reasoning tools. The implementation was based on the theorem provers Spass (Weidenbach 2001), E (Schulz 2004), and Prover9 (McCune 2005–2010), and on the model builder Mace4 (McCune 2005–2010). The provers and the model builder were assigned a maximum of 30 seconds to work on each problem, and were terminated if this was insufficient to find a result.

Two experiments were conducted, one with the strong axiomatization $\mathscr{A}^{s}(p \rightarrow \gamma)$ and one with the weak axiomatization $\mathscr{A}^{w}(p \rightarrow \gamma)$. Since finite model building techniques are restricted to finite models and the strong axiomatization has only infinite models, the experiment with the strong axiomatization was run with the theorem provers alone. Only those meaning postulates α_i were included in the premise term p which belonged to constants occurring in the translation of the input. It is to be expected that meaning postulates unrelated to the input will usually not be relevant, at the same time, their presence would likely slow down the search more than they would help.

Since a term that is a $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-tautology for some $\mathscr{C}$ is also a Henkin tautology and a tautology in the standard sense, finding a proof with the strong or the weak translation guarantees that an inference is valid. Model building behaves differently: finding a model for the weak translation only guarantees satisfiability in a weak notion of semantics, but not satisfiability in a general $\mathscr{L}_{\mathrm{Ty2}}$-model, much less satisfiability in a full model. Applied to weak translations, our inference engine may therefore deem invalid an inference that would in fact be valid under standard semantics. As mentioned above, our test suite contains only one item whose treatment requires the strong axiomatization. For all other items, the predictions are the same for the semantics underlying the weak translation and for standard semantics.

4.5 *Results*

The overall success rates of the provers on valid and contradictory items in the test suite are summarized in Table 2. The figure in the column 'Some' expresses the percentage of items for which at least one prover found a proof or refutation. The 'strong' row shows results for the Henkin-complete axiomatization $\mathscr{A}^s$, the 'weak' row for the translation with the weakened axiomatization $\mathscr{A}^w$.

Table 3 shows the percentage of proofs found within a certain time interval, ranging from 0.1 up to 5 seconds.

There was no test item for which a proof was found under the strong axiomatization but not under the weak axiomatization. In other words, for our test items no proofs were lost by weakening the axiomatization. The performance of the model builder is summarized in Table 4.

Table 2:
Success rates of provers

	Spass	Prover9	E	Some
strong	24.6%	47.7%	23.1%	50.8%
weak	53.8%	76.9%	38.5%	87.7%

	≤ 0.1 s	≤ 1 s	≤ 2 s	≤ 5 s
strong	24.2%	79%	83.9%	91.9%
weak	45.5%	74.5%	82.7%	94.5%

Table 3: Time required by provers

Recall (models found where expected)	78.7%
Accuracy (models expected where found)	98.5%
Determined items (among contingent ones)	56.6%
% of models found within 0.1 seconds	91.1%
% of models found within a second	96.3%
% of models found within two seconds	96.3%
% of models found within five seconds	97.8%

Table 4: Performance of model builder

		Total	Sp	P9	E	Mace4	Success
first-order	valid	4	2	4	4	3/0/0	4
	contradictory	1	1	1	1	0/1/0	1
	contingent	1	0	0	0	1/1/1	1
modality	valid	8	7	8	7	7/0/0	8
	contingent	7	0	0	0	7/4/4	4
knowledge/ belief	valid	6	5	5	2	6/0/0	6
	contingent	4	0	0	0	4/3/3	3
quantifiers	valid	24	16	13	1	18/1/1	18
	contradictory	4	1	1	0	1/3/1	2
	contingent	31	0	0	0	27/16/16	16
adjectives	valid	10	3	10	4	10/0/0	10
	contradictory	7	0	7	5	0/5/0	7
	contingent	8	0	0	0	6/8/6	6
de dicto	valid	1	0	1	1	1/0/0	1
	contingent	1	0	0	0	1/1/1	1
(total)		117	35	50	25	92/43/33	88

Table 5: Results for the weak translation. For the theorem provers, the figures are the numbers of items proven. For Mace4, the figures are the number of items such that a model was found (a) for $\mathscr{A} \cup \{T(p \wedge \gamma)\}$, (b) for $\mathscr{A} \cup \{T(p \wedge \neg\gamma)\}$, and (c) for both problems

Combining the information obtained from the model builder with the results from the provers, all components of our inference engine together provide enough information to determine whether an item is valid, contingent, or contradictory in 75.2% of the cases. The success rates of each theorem prover and the model builder are summarized in Table 5, organized by the semantic phenomena that structure the test suite (as depicted in detail in Appendix D).

4.6 Discussion

The lower success rates for the strong axiomatization indicate that the additional axioms make automated inference harder when they are not relevant to proving. This effect has often been observed when automated deduction is applied to large axiom sets (e.g., Hoder and Voronkov 2011). The additional strength does not provide an advantage in the context of our test suite, as only one item depends on it, and for that item a proof is not found even with the strong axiomatization. The predictions with respect to $\mathscr{A}^w$ and the Henkin-complete $\mathscr{A}^s$ agree on all other items. With this general result in mind, we will focus our discussion on the results obtained with the weak axiomatization.

The difficulty of the test items for reasoning varied with the linguistic phenomena. The combined performance of the reasoning engines on items dealing with first-order tautologies, modality, knowledge and belief, and adjectives is satisfactory, with a success rate of 89.3%. It is unclear why Spass and E perform rather poorly on the section on adjectives, while Prover9 proves all items.

The section on generalized quantifiers was clearly much harder for the systems and reveals the limitations of the current approach. Among the validities and contradictions in that section, eight items (28.6%) remain undetermined. Two of the undetermined items are statements about monotonicity. While it is not clear why item (3.23) (monotonicity of ‘at least three’) is not proved, items (3.15) (‘Most women dance. ⊢ Some women dance.’) and (3.22) (‘Most men dance and play air guitar. ⊢ Most men dance.’) express properties of ‘most’ that are probably too hard to prove automatically on the basis of (1). It is not clear why the provers did not succeed on the items derived from FraCaS.

Two items, (3.2) and (3.14), are wrongly classified as contingent because the proof requires the meaning postulate (24c) for ‘few’,[20] which is not included in the input, as ‘few’ does not occur in the items in question. It is noticeable that among the contingent items, models

[20] Both of these items require the inference that MANY(P,Q) cannot hold when $P \cap Q = \emptyset$. In the context of our meaning postulates, this follows from the mutual exclusiveness of FEW and MANY (see (24b)) and the monotonicity of MANY (see (24a-ii)) when also considering that FEW(P,Q) holds whenever $P \cap Q = \emptyset$ (see (24c)).

verifying the conjecture are found more often than models falsifying it, resulting in only 56.6% of them being determined. The reason might be that models falsifying these inferences are often necessarily larger than the smallest models verifying it. For instance, a model verifying the implication ‘Few blond men dance’ ⇒ ‘Few men dance’ need only contain a single element of type e, but to show that ‘Few blond men dance’ does not generally entail ‘Few men dance’ (item (3.36)), one needs at least two objects and also a function $f \in D_{ee}$.[21]

With respect to our axiomatization, the quantifiers and determiners fall into three classes: those for which we have given a direct definition (*most, at most two, at least three*), those which are indirectly characterized in terms of their properties (*few*, *many*, *several*), and the definite article, which is directly translated as the ι operator. As the indirect characterizations involve direct statements about monotonicity and conservativity, which are targeted by most test items, it is not surprising that the inference engines perform better on *few*, *many*, and *several* than on *most*. The first-order-definable quantifiers *at most two* and *at least three* show a success rate comparable to the other quantifiers. This difference is not surprising, either: the definition of *most* is more complex than the definitions of the numerical quantifiers, and the test items on ‘most’ require that the provers make inferences that are equivalent to proving statements such as Proposition 7 (upward monotonicity in the second argument). Our observations on the success rates of the provers simply emphasize the point that the way in which meaning postulates are stated may have a significant influence on the feasibility of automated inference. Nonetheless, the success of the model builder Mace4 on item (3.29) (‘Most men dance. $\ltimes$ Most men dance and play air guitar.’) demonstrates that the direct definition of *most* can in principle be useful for automated reasoning.

5 RELATED WORK

Higher-order reasoning with natural language is not a lively research area, but there are a number of related fields. In this section we discuss an alternative recent system for reasoning with quantifiers, and earlier work with higher-order provers and higher-order model building.

[21] Since $[\![man(w)]\!] \supseteq [\![blond(w, man)]\!]$, $[\![man(w)]\!]$ cannot be equal to $[\![blond(w, man)]\!]$ when the implication is false.

The Natlog system (MacCartney 2009) is in some respects closest to some of the targets of our reasoning architecture, and it was also evaluated on the basis of the FraCaS test suite. By aligning premises with conjectures at the word level, computing entailment relations between them, and deriving the entailment relation between the sentences by projecting the individual entailment relations using a syntactic dependency analysis, Natlog is able to reason with quantifiers and negation. In contrast to our grammar fragment, Natlog has wide coverage, but it cannot handle problems with more than one premise (MacCartney 2009, p. 142), which, as we saw, are unproblematic for approaches based on theorem proving such as ours. On the single-premise problems, which constitute 53% of the test suite, MacCartney reports an accuracy of 70.5%, with 89.3% precision and 65.7% recall on the binary task of recognizing valid inferences. On the section on adjectives, our experiments show an accuracy of 66.7%, with 86.6% precision and 75% recall (relative to the original FraCaS annotation). These results are comparable to MacCartney's figures for the same section: 71.4% accuracy, 83.3% precision, and 80% recall. However, both our system and Natlog only covered 15 items from this section; they only intersect on 11 items. The situation is different in the section on generalized quantifiers. The decision rate of our system is 57%, whereas Natlog, by virtue of its architecture, makes some decision on every sentence. With undetermined items taken as 'wrongly classified', our system achieves 28.6% accuracy, 66.7% precision, and 0% recall, since only the model builder had some success on the FraCaS data on generalized quantifiers. These figures are far lower than those of the Natlog system, which are 95.2% accuracy, 100% precision, and 97.7% recall, respectively.

Unfortunately, a quantitative comparison between MacCartney's and our results is not very meaningful overall with the data we have so far, considering that (1) our system as well as Natlog only tested portions of the FraCaS test suite, (2) the intersection between the tested items was even smaller, and (3) the development of Natlog was guided by the FraCaS data whereas the predictions of our model partly deviate from the FraCaS annotation. Although comparing the raw numbers produced by Natlog with our system's performance clearly indicates that there is much room for improvement in the generalized quantifiers section, our results also suggest that in principle natural language problems of the type encoded in the FraCaS test suite can be solved by theorem proving, provided that a system is given access to appropriate meaning postulates for (classes of) lexical items.

A very exciting competitor to reasoning with a translation to first-order logic arises from automated reasoning tools that work directly on higher-order logic. Ramsay (1995) presents a special automatic proof system for an intensional logic, based on the property theory of Turner (1987). Kohlhase and Konrad (1998) apply the higher-order theorem prover HOT to corrections in natural language, using the higher-order unification analysis proposed by Dalrymple *et al.* (1991).[22] The difference in the application domain and the considerable advances in automated theorem proving in the last 15 years makes this older work hard to compare to our present study. In order to see what higher-order reasoning can achieve and how it compares to translations under Henkin semantics, it would be interesting to observe the performance of more recent higher-order theorem provers such as LEO II (Benzmüller *et al.* 2007) or Satallax (Brown 2013) with Ty2 representations that result from parsing natural language. We leave such comparison for future work.

Another interesting perspective on higher-order reasoning is provided by investigating the potential of model builders for natural language. Konrad (2004) presents the higher-order model builder *Kimba* and puts it to the test with linguistic data. In particular he uses it to determine the referent of definites within a discourse and to find the valid readings of sentences involving reciprocals. Konrad develops a model builder for a fragment of higher-order logic whose design is guided by typical properties of representations for natural language. In our reasoning architecture, first-order model generation comes after the weak version of the translation from Ty2 representations, and our approach to model-building targets full Ty2, which of course comprises a large class of expressions which are irrelevant for natural language semantics. A further notable difference concerns our treatment of generalized quantifiers such as ‘most’, which turns out to be complementary to what Konrad did. Recall that we define MOST by a meaning postulate within the representation language. Konrad de-

[22] They use HOT to prove that, for instance, ‘No, PETER likes Mary’ is a valid response to ‘Jon likes Mary’, while ‘No, PETER likes Sarah’ is not.

fines it by means of MORE, whose interpretation is directly fixed by a model-theoretic constraint. It seems plausible that defining functions such as MOST model-theoretically rather than by meaning postulates can make model generation vastly more efficient. In particular, numerical quantifiers like ‘two’ and ‘three’ have complex definitions in $\mathscr{L}_{\mathrm{Ty2}}$, which soon become completely intractable as their size grows with the number that they encode. Such quantifiers are far more naturally defined model-theoretically. While model-theoretic definitions have their limitations in the context of proof systems (as we argued in Section 4.2), they fit very naturally into systems for finite model generation: a reasonable model-theoretic definition will in general be decidable on finite structures. Conversely, it would be interesting to explore whether extending our fragment could lead to a successful application of this type of architecture to Konrad’s data. Of particular interest would be a treatment of plurals, reciprocals, and definites. For the latter we assumed a simplistic analysis with a choice operator, and plural did not receive any treatment at all, although it is clearly highly relevant for a more realistic and comprehensive coverage of naturally occurring data.

6 CONCLUSION

We defined and discussed translations under Henkin semantics from Ty2 to first-order logic for automated reasoning with natural language, and investigated the performance of a reasoning architecture with several first-order theorem provers and a model generator on a test suite targeting typical reasoning tasks of theoretical semantics. Unlike previous work on automated reasoning with natural language, we took as input formulae in higher-order logic as proposed by formal semanticists. The architecture was evaluated on a set of 117 natural language inference problems, partly derived from the classical FraCaS test suite originally compiled for such purposes. The inference tasks were expressed in a small fragment of English; they focused on modality, propositional attitudes, generalized quantifiers, and adjectives, and relied on a set of associated meaning postulates commonly assumed in semantics.

The results are promising: Despite the general undecidability of first-order logic, 75.2% of the test items could be determined, a great majority in less than a second. The success rate of the combined inference engines suggests that theorem proving with higher-order representations for natural-language expressions can indeed be reduced to first-order proving by adopting Henkin semantics. At the same time, the system's poor performance on generalized quantifiers confirms expectations that the syntactic form of meaning postulates plays a significant role in the efficiency and ultimate success (or failure) of automated inference. Fine-tuning of meaning postulates and finding a good balance in exploiting the complementary strengths of theorem provers and model builders will be necessary to improve performance.

Our experiments enlisted model builders only in combination with a weak translation, which makes model generation unsound relative to stronger versions of Henkin semantics. Unsoundness did not affect any items in our particular test suite, but we need a better understanding of which classes of terms occurring in logical translations of natural language may cause trouble. In addition, the models created by the model builder suffer from being virtually incomprehensible to human readers due to their compact encoding of functions and types. Readability and usefulness of the models could be greatly enhanced by disentangling these structures automatically for human exploration.

Determining the precise advantages and disadvantages of translations of different strength remains a general desideratum. First-order generation for stronger translations is not generally impossible, but it must be prepared to cope with the fact that general models of higher-order logic are always infinite because the set of types is infinite. Advanced techniques for generating and representing infinite models, such as the ones introduced by Caferra *et al.* (2004), might offer viable solutions. At the other end of the scale, it is also interesting to explore which weakenings of the translation are best suited to natural language applications, and to exploit the advantages of smaller structures. This strategy has to take into account the dangers of potentially unsound translations. Our results suggest that the strength of our weak axiomatization is a promising choice as long as the meaning postulates are chosen carefully.

The setting in which we tested the feasibility of first-order translations of Ty2 in automated reasoning was restricted to a toy grammar and to test cases that belong to the theoretical toolbox of formal semanticists. Opening up its application to broad coverage and to accounting for effects of discourse pragmatics and world knowledge would of course expose the usual weaknesses of deductive reasoning when confronted with potentially incomplete knowledge on the one hand and an overwhelming amount of relevant facts on the other hand (see the discussion in Ovchinnikova 2012, pp. 73–92). However, despite the considerable challenges ahead, Bos (2006) and Bos and Markert (2006) report that automated inference on first-order representations of natural language can succeed in real-world applications. If this is correct, future work should investigate to what extent our translations can successfully widen the empirical scope of Bos' work to encompass semantic effects of intensionality and generalized quantifiers, replacing hand-encoded first-order approximations with well-studied higher-order analyses. The DRT-based wide-coverage Boxer system (Bos 2008) seems a promising starting point to extending the linguistic coverage.

ACKNOWLEDGMENTS

We are indebted to the three reviewers for their valuable comments, and to Johannes Dellert for extensive discussion and comments that had a guiding influence on this work. Janina Radó mastered the challenge of proofreading the paper and suggested numerous improvements.

APPENDICES

A AXIOMS

All axioms with type parameters are shown in a form with type constants as described for the weak axiomatization. The versions with quantification over types introduced in Section 3.2 (for the strong axiomatization) are derived straightforwardly.

First group:

Axioms for Typing and the Axioms of Extensionality

(31) Typing

a. $\forall x_0 : \mathit{isTrue}(x_0) \rightarrow \mathit{hasType}(x_0, t)$

b. $\forall x_0 \forall x_1 \forall x_2 : [\exists x_3 \mathit{hasType}(x_0, g(x_3, x_2)) \wedge \mathit{hasType}(x_1, x_3)]$
$\rightarrow \mathit{hasType}(f(x_0, x_1), x_2))$

c. $\exists x : \mathit{hasType}(x, e) \wedge \exists y : \mathit{hasType}(y, s)$

(32) Axioms of Extensionality

a. $\forall x_0 \forall x_1 \forall x_2 \forall x_3 : [\mathit{hasType}(x_0, g(x_2, x_3)) \wedge$
$\mathit{hasType}(x_1, g(x_2, x_3))]$
$\rightarrow [\forall x_4 \mathit{hasType}(x_4, x_2) \rightarrow f(x_0, x_4) = f(x_1, x_4)] \rightarrow x_0 = x_1$

b. $\forall x_0 \forall x_1 : [[\mathit{hasType}(x_0, t) \wedge \mathit{hasType}(x_1, t)]$
$\wedge [\mathit{isTrue}(x_0) \longleftrightarrow \mathit{isTrue}(x_1)]] \rightarrow x_0 = x_1$

Second group:

Defining Axioms for Ty2 Constants instantiated for all types ρ, σ, τ:

(33) For every constant c^n_τ:
$\mathit{hasType}(T_{term}(c^n_\tau), T_{ty}(\tau))$

(34) For every variable x^n_σ:
$\mathit{hasType}(x^n, T_{ty}(\sigma))$

(35) Combinators

a. $\forall x_0 : \mathit{hasType}(x_0, T_{ty}(\sigma)) \rightarrow f(\mathbf{I}(\sigma), x_0) = x_0$

b. $\forall x_0 \forall x_1 : [\mathit{hasType}(x_0, T_{ty}(\sigma)) \wedge \mathit{hasType}(x_1, T_{ty}(\tau))]$
$\rightarrow f(f(\mathbf{K}(T_{ty}(\sigma), T_{ty}(\tau)), x_0), x_1) = x_0$

c. $\forall x_0 \forall x_1 \forall x_2 : [\mathit{hasType}(x_0, g(T_{ty}(\tau), g(T_{ty}(\sigma), T_{ty}(\rho))))$
$\wedge \mathit{hasType}(x_1, g(T_{ty}(\tau), T_{ty}(\sigma))) \wedge \mathit{hasType}(x_2, T_{ty}(\tau))]$
$\rightarrow f(f(f(\mathbf{S}(T_{ty}(\tau), T_{ty}(\sigma), T_{ty}(\rho)), x_0), x_1), x_2)$
$= f(f(x_0, x_2), f(x_1, x_2))$

(36) Equality

a. $\forall x_1 \forall x_2 : \mathit{isTrue}(f(f(\dot{\equiv}(T_{ty}(\sigma)), x_1), x_2))$
$\longleftrightarrow (\mathit{hasType}(x_1, T_{ty}(\sigma)) \wedge x_1 = x_2)$

(37) Choice

a. $\forall x_1 : \mathit{hasType}(x_1, g(T_{ty}(\sigma), t)) \rightarrow [\exists x_2 \mathit{isTrue}(f(x_1, x_2)))$
$\rightarrow \mathit{isTrue}(f(x_1, f(\iota(T_{ty}(\sigma)), x_1)))]$

(38) Existential Quantifier

a. $\forall x_1 \mathit{hasType}(x_1, T_{ty}(\sigma)) \rightarrow [\forall x_2 : \mathit{isTrue}(f(\dot{\exists}(T_{ty}(\sigma)), x_2))$
$\longleftrightarrow \exists x_3(\mathit{hasType}(x_3, T_{ty}(\sigma)) \wedge \mathit{isTrue}(f(x_2, x_3))]$

(39) Universal Quantifier

a. $\forall x_1 \mathit{hasType}(x_1, T_{ty}(\sigma)) \rightarrow [\forall x_2 : \mathit{isTrue}(f(\dot{\forall}(T_{ty}(\sigma)), x_2))$
$\longleftrightarrow \forall x_3(\mathit{hasType}(x_3, T_{ty}(\sigma)) \rightarrow \mathit{isTrue}(f(x_2, x_3)))]$

(40) Propositional Connectives

a. $\forall x_0 : \mathit{hasType}(x_0, t) \rightarrow$
$(\mathit{isTrue}(f(\dot{\neg}, x_0)) \leftrightarrow \mathit{not}(\mathit{isTrue}(x_0))))$

b. $\forall x_0 \forall x_1 : [\mathit{hasType}(x_0, t) \wedge \mathit{hasType}(x_1, t)]$
$\rightarrow [\mathit{isTrue}(f(f(\dot{\wedge}, x_0), x_1)) \leftrightarrow (\mathit{isTrue}(x_0) \wedge \mathit{isTrue}(x_1))]$

c. $\forall x_0 \forall x_1 : [\mathit{hasType}(x_0, t) \wedge \mathit{hasType}(x_1, t)]$
$\rightarrow [\mathit{isTrue}(f(f(\dot{\rightarrow}, x_0), x_1)) \leftrightarrow (\mathit{isTrue}(x_0) \rightarrow \mathit{isTrue}(x_1))]$

d. $\forall x_0 \forall x_1 : [\mathit{hasType}(x_0, t) \wedge \mathit{hasType}(x_1, t)]$
$\rightarrow (\mathit{isTrue}(f(f(\dot{\vee}, x_0), x_1)) \leftrightarrow (\mathit{isTrue}(x_0) \vee \mathit{isTrue}(x_1)))$

(41) Function symbols for lambda abstracts: see (10)

B PROOFS

B.1 *Soundness and completeness*

In this section, we prove Theorem 8. We adapt familiar proofs of Henkin's completeness theorem based on first-order translations and the first-order completeness theorem (van Benthem and Doets 1983, pp. 276–283, Leivant 1994, sections 5.4–5.5). First we embed $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$ in a multi-sorted first-order language, $\mathscr{F}^{\mathscr{C}}$:

Definition 10. *The sorts of $\mathscr{F}^{\mathscr{C}}$ are the types of $\mathscr{L}_{\mathrm{Ty2}}$. The terms of $\mathscr{F}^{\mathscr{C}}$ are the terms of $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$ plus the variables of $\mathscr{L}_{\mathrm{Ty2}}$, and the sort of α_σ as a term of $\mathscr{F}^{\mathscr{C}}$ is σ. The variables of $\mathscr{F}^{\mathscr{C}}$ are the variables of $\mathscr{L}_{\mathrm{Ty2}}$. A lambda abstract $\lambda x.\alpha$ with n free variables $x^1, \ldots, x^n$ is understood as an n-place function symbol applied to $x^1, \ldots, x^n$. The language $\mathscr{F}^{\mathscr{C}}$ of $\mathscr{C}$-formulae is the smallest set such that*

- *$\mathit{isTrue}(\alpha_t) \in \mathscr{F}^{\mathscr{C}}$ for every term $\alpha_t \in \mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$ of sort t. We will write α_t for $\mathit{isTrue}(\alpha_t)$.*
- *If α, $\beta \in \mathscr{F}^{\mathscr{C}}$, then $(\alpha \hat{\circ} \beta) \in \mathscr{F}^{\mathscr{C}}$ for all propositional connectives $\circ$, similarly for negation*
- *If $\alpha \in \mathscr{F}^{\mathscr{C}}$ and x_σ is a variable, then $(\hat{\forall}^\sigma x_\sigma\ \alpha) \in \mathscr{F}^{\mathscr{C}}$ and $(\hat{\exists}^\sigma x_\sigma\ \alpha) \in \mathscr{F}^{\mathscr{C}}$*
- *If $\alpha_\sigma, \beta_\sigma \in \mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$, then $(\alpha_\sigma \hat{\equiv}^\sigma \beta_\sigma) \in \mathscr{F}^{\mathscr{C}}$*

We interpret $\mathscr{F}^{\mathscr{C}}$ over structures in which the universes may be empty for some sorts and where therefore the interpretation of a term or formula *with free variables may be undefined. An $\mathscr{F}^{\mathscr{C}}$-structure $\mathfrak{M}$ has as its universe a family $\{D_\tau : \tau \in Types\}$ of mutually disjoint sets, where $D_s, D_e \neq \emptyset$, and for all other types τ, D_τ is non-empty (at least) if there is a term of type τ in $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$, interpreting constants of sort σ by elements of D_σ. $\mathscr{F}^{\mathscr{C}}$-structures interpret $\equiv^\sigma$ as equality between objects of sort σ, and provide an interpretation for the predicate symbol $isTrue(\cdot_t)$, the function symbol $(\cdot_{\langle\tau\sigma\rangle}, \cdot_\tau)_\sigma$ for each pair of sorts τ, $\sigma \in Types$, and all function symbols representing lambda abstracts $\lambda x.\alpha \in \mathscr{C}$. An $\mathfrak{M}$-assignment v is a partial function from the variables of $\mathscr{L}_{\mathrm{Ty2}}$ such that $v(x_\sigma) \in D_\sigma$ for every x_σ in the domain of v. Evaluation is defined as follows:*

- *$[\![\alpha]\!]^v_{\mathfrak{M}}$ is undefined if v is undefined for some variable occurring free in α.*
  Otherwise, we have:
- $[\![x]\!]^v_{\mathfrak{M}} = v(x)$
- $[\![(\alpha_{\langle\tau\sigma\rangle}\beta_\tau)]\!]^v_{\mathfrak{M}} = [\![(\cdot_{\langle\tau\sigma\rangle}, \cdot_\tau)_\sigma]\!]_{\mathfrak{M}}([\![\alpha]\!]^v_{\mathfrak{M}}, [\![\beta]\!]^v_{\mathfrak{M}})$
- $[\![(\hat{\forall}^\sigma x_\sigma\ \alpha)]\!]^v_{\mathfrak{M}} = 1$ *iff* $[\![\alpha]\!]^{v'}_{\mathfrak{M}} = 1$ *for every $\mathfrak{M}$-assignment v' that has x_σ in its domain and that agrees with v on $Domain(v)\backslash\{x_\sigma\}$, and 0 otherwise*
- $[\![(\hat{\exists}^\sigma x_\sigma\ \alpha)]\!]^v_{\mathfrak{M}} = 1$ *iff* $[\![\alpha]\!]^{v'}_{\mathfrak{M}} = 1$ *for some $\mathfrak{M}$-assignment v' that has x_σ in its domain and that agrees with v on $Domain(v)\backslash\{x_\sigma\}$, and 0 otherwise*

with straightforward clauses for atoms and propositional connectives. A structure $\mathfrak{M}$ verifies a formula $\phi \in \mathscr{F}^{\mathscr{C}}$ iff $[\![\phi]\!]^v_{\mathfrak{M}} = 1$ for all $\mathfrak{M}$-assignments v that have all free variables of ϕ in their domain.

There is a canonical translation from $\mathscr{F}^{\mathscr{C}}$ to $\mathscr{L}_{\mathrm{Ty2}}$, but $\mathscr{F}^{\mathscr{C}}$ is in general more expressive than $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$. The formula translation T_f can be extended canonically to $\mathscr{F}^{\mathscr{C}}$. Axioms from $\mathscr{A}^{\mathscr{C}}$ that do not contain quantification over types can be understood as formulae of $\mathscr{F}^{\mathscr{C}}$. For instance, the first-order formula representing the Axiom of Extensionality for objects of type t, (32b), can be identified with the $\mathscr{C}$-formula

(42) $\hat{\forall}^t x_t \hat{\forall}^t y_t : (x_t \hat{\leftrightarrow} y_t) \hat{\rightarrow} (x_t \hat{\equiv}^t y_t)$.

Furthermore, every axiom that does not contain positive occurrences of variables representing types can be associated with a (possibly infinite) family of $\mathscr{C}$-formulae. For instance the first-order formula of the Axiom of Extensionality for higher types, (32a), can be associated with the set

(43) $\{\hat{\forall}^{\sigma} x_{\sigma}(((\phi_{\sigma\tau} x_{\sigma})\hat{\equiv}^{\tau}(\psi_{\sigma\tau} x_{\sigma})))\hat{\rightarrow}((\phi \hat{\equiv}^{\sigma\tau} \psi) : \sigma, \tau \in \mathit{Types}\}$.

The only axioms from $\mathscr{A}^{\mathscr{C}}$ containing positive occurrences of variables representing types are the typing axioms (31, 33, 34). As these axioms only fix the types of the constant symbols, they are already implicit in the syntax of $\mathscr{F}^{\mathscr{C}}$. Replacing the others by $\mathscr{C}$-formulae in this manner, we obtain a set $\mathscr{B}^{\mathscr{C}} \subset \mathscr{F}^{\mathscr{C}}$ of $\mathscr{C}$-formulae representing all axioms in $\mathscr{A}^{\mathscr{C}}$ apart from the typing axioms.

The notion of an $\mathscr{L}^{\mathscr{C}}_{Ty2}$-interpretation is straightforwardly extended to $\mathscr{F}^{\mathscr{C}}$, and $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-models can therefore be canonically viewed as $\mathscr{F}^{\mathscr{C}}$-structures. We obtain the following characterization:

Proposition 11. *The class of $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-models is (via this identification) equal to the class of $\mathscr{F}^{\mathscr{C}}$-structures that satisfy the formulae in $\mathscr{B}^{\mathscr{C}}$.*

Proof. Immediate from Definition 4. □

We now proceed to the proof of completeness (Lemma 12) and soundness (Lemma 13).

Lemma 12. *If $\phi \in \mathscr{F}^{\mathscr{C}}$ is true in all $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-models, then $\mathscr{A}^{\mathscr{C}} \vdash T(\phi)$.*

Proof. We show that if $T(\phi)$ is satisfiable in a model of $\mathscr{A}^{\mathscr{C}}$, then ϕ is true in some $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-model. The claim then follows from the completeness theorem for first-order logic.

Let $\mathfrak{M}$ be a model of $\mathscr{A}^{\mathscr{C}}$ that verifies $T(\phi)$. For every $\alpha \in \mathfrak{M}$ and every type σ such that $[\![\mathit{hasType}]\!]_{\mathfrak{M}}(\alpha, [\![T_{ty}(\sigma)]\!]_{\mathfrak{M}}) = T$, take a fresh object $\overline{\alpha}_{\sigma}$. Set $\overline{\alpha}_{\sigma\tau}(\overline{\beta}_{\sigma})$ to be $\overline{[\![f]\!]_{\mathfrak{M}}(\alpha, \beta)}_{\tau}$, which exists by the second typing axiom. By the Axiom of Extensionality, there can be at most two objects of type t. Identify the one that verifies *isTrue*, if it exists, with T, and the other one, if it exists, with F. We thus obtain a frame $\{D_{\sigma} : \sigma \in \mathit{Types}\}$, from which we build an $\mathscr{F}^{\mathscr{C}}$-structure $\mathfrak{M}'$ by setting $[\![c]\!]_{\mathfrak{M}'}$ to $[\![c]\!]_{\mathfrak{M}}$ for every constant $c \in \mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$. In the case that c is a polymorphic logical constant $c^{\tau_1,\ldots,\tau_n}$, i.e., a quantifier or a combinator, we set $[\![c^{\tau_1,\ldots,\tau_n}]\!]_{\mathfrak{M}'}$ to $[\![c]\!]_{\mathfrak{M}}([\![T(\tau_1)]\!]_{\mathfrak{M}}, \ldots, [\![T(\tau_n)]\!]_{\mathfrak{M}})$ for all types $\tau_1, \ldots, \tau_n$.

We then need to show $\mathfrak{M} \models T(\psi) \Rightarrow \mathfrak{M}' \models \psi$ for all $\psi \in \mathscr{F}^{\mathscr{C}}$. First, by induction, $[\![t]\!]^{v}_{\mathfrak{M}'} = [\![T_{term}(t)]\!]^{w}_{\mathfrak{M}}$ for every term $t \in \mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$ and all assignments $v\colon Var \cap \mathscr{L}^{\mathscr{C}}_{Ty2} \rightarrow \mathfrak{M}'$, $w\colon Var \rightarrow \mathfrak{M}$ such that $w \supset v$, where *Var* is the set of Ty2 variables. The claim follows by induction over formula structure. The structure $\mathfrak{M}$ verifies the first-order translation of every element of $\mathscr{B}^{\mathscr{C}}$. Therefore, by Proposition 11, $\mathfrak{M}'$ is a $\mathscr{L}^{\mathscr{C}}_{Ty2}$-model and, in particular, ϕ has a $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-model. □

We have shown that every model of $\mathscr{A}^{\mathscr{C}}$ encodes a $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-model, preserving the truth of $\mathscr{F}^{\mathscr{C}}$-formulae. This shows that a finite model $\mathfrak{M}$ of a first-order translation generated by a model builder can be viewed as encoding a (possibly infinite) $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-model such that the value of every $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-term in this model can be computed from $\mathfrak{M}$.

Lemma 13. *Let $\phi \in \mathscr{F}^{\mathscr{C}}$. If $\mathscr{A}^{\mathscr{C}} \vdash T(\phi)$, then ϕ holds in all $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-models.*

Proof. Assume $\neg\phi$ holds in some $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-model $\mathfrak{M}$. As above, we interpret $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-models as multi-sorted first-order structures. We extend them by adding every $\tau \in$ *Types* to the universe, giving them a separate sort, and defining predicates *isTrue* and $\equiv^{\sigma}$ straightforwardly. Then we can interpret the first-order language of the translation in these structures. Obtain a first-order structure $\mathfrak{M}'$ in this manner. We show that $\mathfrak{M} \models \psi \Rightarrow \mathfrak{M}' \models T(\psi)$ for all $\psi \in \mathscr{F}^{\mathscr{C}}$. Thus $\mathfrak{M}' \models T(\neg\phi)$. The typing axioms are evidently true in $\mathfrak{M}'$. The other axioms are true in $\mathfrak{M}'$ by Proposition 11. Thus $\mathscr{A}^{\mathscr{C}} \not\models T(\phi)$, and by soundness of first-order deduction, $\mathscr{A}^{\mathscr{C}} \nvdash T(\phi)$. □

This concludes the proof of Theorem 8.

B.2 *Model building*

In this section, we prove Theorem 9.

Proof. Set $\mathscr{C} := \{\alpha\}$. Let $\langle D, \mathscr{I}\rangle$ be a general $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-model such that D_e and D_s are finite, and α_t is true in $\langle D, \mathscr{I}\rangle$. We say that e, t, s have rank 1, and the rank of $g(\sigma, \tau)$ is one plus the maximum of the ranks of σ and τ. Let n be twice the maximum rank of all the types of sub-terms occurring in α. Obtain a finite $\mathscr{L}^{\mathscr{C}}_{\mathrm{Ty2}}$-model $\mathfrak{N}$ by setting $D_{\tau} := \emptyset$ for all types τ of rank $> n$. As in the proof of Lemma 13, we can view $\mathfrak{N}$ as a first-order-structure that verifies $\mathscr{A}^{\mathscr{C}} \wedge T(\alpha)$. □

C MEANING POSTULATES

For every postulate, the set of triggering constant symbols is given. Where it occurs, α stands for the triggering constant symbol.

1. Intersective adjectives: *blond, Scandinavian, Irish, British, female, male*
   $\exists P^1_{\langle s\langle et\rangle\rangle}\forall w_s\forall P^2_{\langle s\langle et\rangle\rangle}\forall x_e(\alpha(w,P^2,x)\longleftrightarrow(P^1(w,x)\wedge P^2(w,x)))$
2. Subsective, non-intersective adjectives: *genuine, skillful, successful, interesting, large, small, fat, tall, blue*
   $\forall P_{\langle s\langle et\rangle\rangle}\forall x_e\forall w_s(\alpha(w,P,x)\rightarrow P(w,x))$
3. Privative adjectives: *fake, former*
   $\forall P_{\langle s\langle et\rangle\rangle}\forall x_e\forall w_s(\alpha(w,P,x)\rightarrow\neg P(w,x))$
4. *alleged*
   $\forall P_{\langle s\langle et\rangle\rangle}\forall x_e\forall w^1_s(\textit{alleged}(w^1,P,x)\longleftrightarrow\textit{allegedly}(w^1,(\lambda w^2P(w^2,x))))$
   Note that this axiom is slightly different from the one given in the text (see (21)), but the version here is sufficient for the relevant test items.
5. Mutual exclusiveness of *small, large*
   $\forall w_s\forall x_e\forall P_{\langle s\langle et\rangle\rangle}(\textit{small}(w,P,x)\rightarrow\neg\textit{large}(w,P,x))$
6. *necessarily*
   $\forall w^1_s\forall P_{\langle st\rangle}(\textit{necessarily}(w^1,P)\longleftrightarrow\forall w^2_sP(w^2))$
7. *possibly*
   $\forall w^1_s\forall P_{\langle st\rangle}(\textit{possibly}(w^1,P)\longleftrightarrow\exists w^2_sP(w^2))$
8. *two*
   $\forall P^1_{\langle et\rangle}\forall P^2_{\langle et\rangle}(two^e(P^1,P^2)\longleftrightarrow\exists x^1_e\exists x^2_e(x^1\not\equiv x^2\wedge(P^1(x^1)\wedge P^1(x^2))))$
9. *at-most-two*
   $\forall P^1_{\langle et\rangle}\forall P^2_{\langle et\rangle}(\textit{at-most-two}^e(P^1,P^2)\longleftrightarrow\exists x^1_e\exists x^2_e\forall x^3_e(x^3\not\equiv x^1\rightarrow(x^3\not\equiv x^2\rightarrow\neg(P^1(x^3)\wedge P^2(x^3)))))$
10. *at-least-three*
    $\forall P^1_{\langle et\rangle}\forall P^2_{\langle et\rangle}(\textit{at-least-three}^e(P^1,P^2)\longleftrightarrow\exists x^1_e\exists x^2_e\exists x^3_e((((((((x^1\not\equiv x^2\wedge x^1\not\equiv x^3)\wedge x^2\not\equiv x^3)\wedge P^1(x^1))\wedge P^2(x^1))\wedge P^1(x^2))\wedge P^2(x^2))\wedge P^1(x^3))\wedge P^2(x^3)))$
11. *most*
    $\forall P^1_{\langle et\rangle}\forall P^2_{\langle et\rangle}(most^e(P^1,P^2)\longleftrightarrow\forall f_{\langle ee\rangle}(\forall x^1_e((P^1(x^1)\wedge\neg P^2(x^1))\rightarrow(P^1(f(x^1))\wedge P^2(f(x^1))))\rightarrow\exists x^2_e((P^1(x^2)\wedge P^2(x^2))\wedge\forall x^3_e((P^1(x^3)\wedge\neg P^2(x^3))\rightarrow f(x^3)\not\equiv x^2))))$

12. *only*
$\forall P^1_{\langle et\rangle}\forall P^2_{\langle et\rangle}(\mathit{only}^e(P^1,P^2) \leftrightarrow \forall x_e(P^2(x) \rightarrow P^1(x)))$
13. Conservativity of SEVERAL
$\forall P^1_{\langle et\rangle}\forall P^2_{\langle et\rangle}(\alpha(P^1,P^2) \rightarrow \exists x_e(P^1(x) \wedge P^2(x)))$
14. Monotonicity: upwards on first argument: SEVERAL, MANY
$\forall P^1_{\langle et\rangle}\forall P^2_{\langle et\rangle}\forall P^3_{\langle et\rangle}(\alpha(P^1,P^2) \rightarrow (\forall x_e(P^1(x) \rightarrow P^3(x)) \rightarrow \alpha(P^3,P^2)))$
15. Monotonicity: upwards on second argument: SEVERAL, MANY
$\forall P^1_{\langle et\rangle}\forall P^2_{\langle et\rangle}\forall P^3_{\langle et\rangle}(\alpha(P^1,P^2) \rightarrow (\forall x_e(P^2(x) \rightarrow P^3(x)) \rightarrow \alpha(P^1,P^3)))$
16. Monotonicity: downwards on first argument: FEW
$\forall P^1_{\langle et\rangle}\forall P^2_{\langle et\rangle}\forall P^3_{\langle et\rangle}(\alpha(P^1,P^2) \rightarrow (\forall x_e(P^3(x) \rightarrow P^1(x)) \rightarrow \alpha(P^3,P^2)))$
17. Monotonicity: downwards on second argument: FEW
$\forall P^1_{\langle et\rangle}\forall P^2_{\langle et\rangle}\forall P^3_{\langle et\rangle}(\alpha(P^1,P^2) \rightarrow (\forall x_e(P^3(x) \rightarrow P^2(x)) \rightarrow \alpha(P^1,P^3)))$
18. Non-empty extension: FEW
$\forall P^1_{\langle et\rangle}\forall P^2_{\langle et\rangle}(\forall x_e\neg(P^1(x) \wedge P^2(x)) \rightarrow \mathit{few}^e(P^1,P^2))$
19. Mutual exclusiveness of MANY and FEW
$\forall P^1_{\langle et\rangle}\forall P^2_{\langle et\rangle}\neg(\mathit{many}^e(P^1,P^2) \wedge few^e(P^1,P^2))$
20. Deductivity: *think, know*
$\forall x_e\forall P^1_{\langle st\rangle}\forall P^2_{\langle st\rangle}\forall w^1_s((\alpha(w^1,P^1,x) \wedge \forall w^2_s(P^1(w^2) \rightarrow P^2(w^2)))$
$\rightarrow \alpha(w^1,P^2,x))$
21. Veridicality: *know*
$\forall x_e\forall P_{\langle st\rangle}\forall w_s(\mathit{know}(w,P,x) \rightarrow P(w))$

D TEST SUITE

Each test item consists of premises and a conjecture, which are separated by a symbol which is $\vdash$ for valid items, $\vdash$ NON for contradictory ones, and $\ltimes$ for contingent items. Our notational conventions and terminology are explained in Section 4.3.

First-order inferences

(0.0) $\vdash$ NON Mia dances and does not dance.

(0.1) Mia dances and does not dance. $\vdash$ Mia dances.

(0.2) $\vdash$ Every man dances or does not dance.

(0.3) Mia is a woman. Mia dances. $\vdash$ A woman dances.

(0.4) Mia is a robber. $\ltimes$ Mia is a man.

(0.5) Mia is a woman. Every woman dances. $\vdash$ Mia dances.

Modality

(1.0) Mia dances. ⊢ Mia possibly dances.

(1.1) Mia dances. ⋉ Mia necessarily dances.

(1.2) Mia is a robber. ⋉ Mia allegedly is a robber.

(1.3) Mia necessarily dances. ⊢ Mia dances.

(1.4) Mia possibly dances. ⋉ Mia dances.

(1.5) Mia allegedly is a robber. ⋉ Mia is a robber.

(1.6) Mia necessarily dances. ⊢ Mia possibly dances.

(1.7) Mia possibly dances. ⋉ Mia necessarily dances.

(1.8) Mia does not possibly dance. ⊢ Mia necessarily does not dance.

(1.9) Mia does not dance. ⊢ Mia does not necessarily dance.

(1.10) Mia does not possibly dance. ⊢ Mia does not dance.

(1.11) ⊢ Mia dances or does not necessarily dance.

(1.12) Mia is an alleged robber. ⊢ Mia allegedly is a robber.

(1.13) Mia necessarily is a robber. ⋉ Mia allegedly is a robber.

(1.14) Mia allegedly is a robber. ⋉ Mia possibly is a robber.

Propositional attitudes

(2.0) Mia thinks that John necessarily dances. ⊢ Mia thinks that John dances.

(2.1) John thinks that Mia knows that Vincent dances. ⊢ John thinks that Vincent dances.

(2.2) John thinks that Mia eats several burgers. ⊢ John thinks that Mia eats a burger.

(2.3) John thinks that Mia is a blond woman. ⊢ John thinks that Mia is a woman.

(2.4) John thinks that Mia is an alleged robber. ⋉ John thinks that Mia is a robber.

(2.5) John knows that Mia dances. ⊢ Mia dances.

(2.6) John thinks that Mia dances. ⋉ Mia dances.

(2.7) Mia necessarily knows that Mia dances. ⊢ Mia necessarily dances.

(2.8) Mia knows that John dances. John is the chairman. ⋉ Mia knows that the chairman dances.

(2.9) Mia knows that John saw a unicorn. ⊢ Some unicorn is a unicorn.

(2.10) Mia thinks that John saw a unicorn. ⋉ Some unicorn is a unicorn.

Generalized quantifiers

(3.0) Few women dance. ⊢ NON Many women dance.

(3.1) No women dance. ⊢ Few women dance.

(3.2) No women dance. ⊢ NON Many women dance.

(3.3) Few women dance. ⋉ No women dance.

(3.4) Many women dance. ⋉ All women dance.

(3.5) All women dance. ⋉ Many women dance.

(3.6) Mia eats every burger. Mia eats a burger. ⊢ Mia eats most burgers.

(3.7) Every man dances. A man dances. ⊢ Most men dance.

(3.8) Mia eats a burger. ⋉ Mia eats most burgers.

(3.9) Mia eats most burgers. ⋉ Mia eats all burgers.

(3.10) Only men dance. No woman is a man. ⊢ No woman dances.

(3.11) John is a man. Every man dances. ⊢ The man dances.

(3.12) At least three women dance. ⊢ NON At most two women dance.

(3.13) The man dances. John is a man. ⊢ A man dances.

(3.14) Many women dance. ⊢ Some women dance.

(3.15) Few women dance. ⋉ Some women dance.

(3.16) Several women dance. ⊢ Some women dance.

(3.17) Most women dance. ⊢ Some women dance.

(3.18) At least three women dance. ⊢ Some women dance.

(3.19) At most two women dance. ⋉ Some women dance.

(3.20) The man dances and plays air guitar. ⊢ The man dances.

(3.21) Many men dance and play air guitar. ⊢ Many men dance.

(3.22) Few men dance and play air guitar. ⋉ Few men dance.

(3.23) Several men dance and play air guitar. ⊢ Several men dance.

(3.24) Most men dance and play air guitar. ⊢ Most men dance.

(3.25) At least three men dance and play air guitar. ⊢ At least three men dance.

(3.26) At most two men dance and play air guitar. ⋉ At most two men dance.

(3.27) The man dances. ⋉ The man dances and plays air guitar.

(3.28) Many men dance. ⋉ Many men dance and play air guitar.

(3.29) Few men dance. ⊢ Few men dance and play air guitar.

(3.30) Several men dance. ⋉ Several men dance and play air guitar.

(3.31) Most men dance. ⋉ Most men dance and play air guitar.

(3.32) At least three men dance. ⋉ At least three men dance and play air guitar.

(3.33) At most two men dance. ⊢ At most two men dance and play air guitar.

(3.34) The blond man dances. ⋉ The man dances.

(3.35) Many blond men dance. ⊢ Many men dance.

(3.36) Few blond men dance. ⋉ Few men dance.

(3.37) Several blond men dance. ⊢ Several men dance.

(3.38) Most blond men dance. ⋉ Most men dance.

(3.39) At least three blond men dance. ⊢ At least three men dance.

(3.40) At most two blond men dance. ⋉ At most two men dance.

(3.41) The man dances. ⋉ The blond man dances.

(3.42) Many men dance. ⋉ Many blond men dance.

(3.43) Few men dance. ⊢ Few blond men dance.

(3.44) Several men dance. ⋉ Several blond men dance.

(3.45) Most men dance. ⋉ Most blond men dance.

(3.46) At least three men dance. ⋉ At least three blond men dance.

(3.47) At most three men dance. ⊢ At most three blond men dance.

Generalized quantifiers (from FraCaS)

(F22) No delegate finished the report on time. ⋉ No delegate finished the report.

(F23) Some delegates finished the survey on time. ⋉ Some delegates finished the survey.

(F24) Many delegates obtained interesting results from the survey. ⊢ Many delegates obtained results from the survey.

(F38) No delegate finished the report. ⋉ Some delegate finished the report on time.

(F39) Some delegates finished the survey. ⋉ Some delegates finished the survey on time.

(F40) Many delegates obtained results from the survey. ⋉ Many delegates obtained interesting results from the survey.

(F54) No Scandinavian delegate finished the report on time. ⋉ Some delegate finished the report on time.

(F55) Some Irish delegates finished the survey on time. ⊢ Some delegates finished the survey on time.

(F56) Many British delegates obtained interesting results from the survey. ⋉ Many delegates obtained interesting results from the survey.

(F63) At least three female commissioners spend time at home. ⊢ At least three commissioners spend time at home.

(F70) No delegate finished the report on time. ⊢ NON Some Scandinavian delegate finished the report on time.

(F71) Some delegates finished the survey on time. ⋉ Some Irish delegates finished the survey on time.

(F72) Many delegates obtained interesting results from the survey. ⋉ Many British delegates obtained interesting results from the survey.

(F79) At least three commissioners spend time at home. ⋉ At least three male commissioners spend time at home.

Adjectives

(4.0) Mia is a blond woman. ⊢ Mia is a woman.

(4.1) Mia is blond. Mia is a woman. ⊢ Mia is a blond woman.

(4.2) Mia is a blond woman. Mia is a robber. ⊢ Mia is a blond robber.

(4.3) Mia has a genuine diamond. ⊢ Mia has a diamond.

(4.4) Mia is a skillful robber. Mia is a boxer. ⋉ Mia is a skillful boxer.

(4.5) Excalibur is a fake sword. ⊢ NON Excalibur is a sword.

(4.6) ⊢ No fake sword is a sword.

(4.7) Excalibur is a weapon. Excalibur is a fake sword. ⊢ NON Excalibur is a fake weapon.

(4.8) Mia is an alleged robber. ⋉ Mia is a robber.

(4.9) Mia is an alleged robber. Mia is a boxer. ⋉ Mia is an alleged boxer.

Adjectives (from FraCaS)

(F197) John has a genuine diamond. ⊢ John has a diamond.

(F198) John is a former university student. ⊢ NON John is a university student.

(F199) John is a successful former university student. ⊢ John is successful.

(F200) John is a former successful university student. ⋉ John is successful.

(F201) John is a former successful university student. ⋉ John is a university student.

(F204) Mickey is a small animal. ⊢ NON Mickey is a large animal.

(F205) Dumbo is a large animal. ⊢ NON Dumbo is a small animal.

(F206) Fido is not a small animal. ⋉ Fido is a large animal.

(F207) Fido is not a large animal. ⋉ Fido is a small animal.

(F210) All mice are small animals. Mickey is a large mouse. ⊢ NON Mickey is a large animal.

(F211) All elephants are large animals. Dumbo is a small elephant. ⊢ NON Dumbo is a small animal.

(F214) All legal authorities are law lecturers. All law lecturers are legal authorities. ⊢ All fat legal authorities are fat law lecturers.

(F215) All legal authorities are law lecturers. All law lecturers are legal authorities. ⋉ All competent legal authorities are competent law lecturers.

(F218) Kim is a clever person. ⊢ Kim is clever.

(F219) Kim is a clever politician. ⊢ Kim is clever.

De dicto

(5.0) Mia seeks a unicorn. ⋉ Some unicorn is a unicorn.

(5.1) Mia sees a unicorn. ⊢ Some unicorn is a unicorn.

REFERENCES

Jon BARWISE and Robin COOPER (1981), Generalized quantifiers and natural language, *Linguistics and Philosophy*, 4(2):159–219.

Michael BENNETT (1974), *Some Extensions of a Montague Fragment*, Ph.D. thesis, UCLA.

Christoph BENZMÜLLER, Larry PAULSON, Frank THEISS, and Arnaud FIETZKE (2007), Progress Report on Leo-II, an Automatic Theorem Prover for Higher-Order Logic, in Klaus SCHNEIDER and Jens BRANDT, editors, *Theorem Proving in Higher Order Logics 2007 – Emerging Trends*, pp. 33–48.

Patrick BLACKBURN and Johan BOS (2005), *Representation and Inference for Natural Language*, CSLI Publications, Stanford.

Johan BOS (2004), Computational semantics in discourse: underspecification, resolution, and inference, *Journal of Logic, Language and Information*, 13:139–157.

Johan BOS (2006), Three stories on automated reasoning for natural language understanding, in *Proceedings of ESCoR (IJCAR Workshop): Empirically Successful Computerized Reasoning*, pp. 81–91.

Johan BOS (2008), Wide-coverage semantic analysis with Boxer, in Johan BOS and Rodolfo DELMONTE, editors, *Semantics in Text Processing. STEP 2008 Conference Proceedings*, Research in Computational Semantics, pp. 277–286, College Publications.

Johan BOS and Katja MARKERT (2006), When logical inference helps determining textual entailment (and when it doesn't), in Bernardo MAGNINI and Ido DAGAN, editors, *The Second PASCAL Recognising Textual Entailment Challenge. Proceedings of the Challenges Workshop*, pp. 98–103, Venice, Italy.

Chad E. BROWN (2013), Reducing higher-order theorem proving to a sequence of SAT problems, *Journal of Automated Reasoning*, 51(1):57–77.

Ricardo CAFERRA, Alexander LEITSCH, and Nicolas PELTIER (2004), *Automated Model Building*, Kluwer.

Alonzo CHURCH (1940), A formulation of the simple theory of types, *The Journal of Symbolic Logic*, 5(2):56–68.

Robin COOPER, Dick CROUCH, Jan VAN EIJCK, Chris FOX, Josef VAN GENABITH, Jan JASPARS, Hans KAMP, David MILWARD, Manfred PINKAL, Massimo POESIO, Steve PULMAN, Ted BRISCOE, Holger MAIER, and Karsten KONRAD (1996), Using the Framework, Technical report, FraCaS Consortium, University of Edinburgh, Deliverable D16 — Final Draft.

Ido DAGAN, Bill DOLAN, Bernardo MAGNINI, and Dan ROTH (2009), Recognizing textual entailment: Rational, evaluation and approaches, *Natural Language Engineering*, 15(4):i–xvii.

Mary DALRYMPLE, Stuart M. SHIEBER, and Fernando C. N. PEREIRA (1991), Ellipsis and higher-order unification, *Linguistics and Philosophy*, 14(4):399–452.

David R. DOWTY, Robert E. WALL, and Stanley PETERS (1981), *Introduction to Montague Semantics*, Reidel, Dordrecht.

Joyce FRIEDMAN and David S. WARREN (1980), λ-normal forms in an intensional model for English, *Studia Logica*, 39:311–324.

Daniel GALLIN (1975), *Intensional and Higher-Order Modal Logic*, North-Holland, Amsterdam.

L.T.F. GAMUT (1991), *Logic, Language and Meaning, Volume II: Intensional Logic and Logical Grammar*, University of Chicago Press.

Jeroen GROENENDIJK and Martin STOKHOF (1982), Semantic analysis of wh-complements, *Linguistics and Philosophy*, 5(2):175–233.

Leon HENKIN (1950), Completeness in the theory of types, *The Journal of Symbolic Logic*, 15(2):81–91.

Klaus VON HEUSINGER (1997), Definite descriptions and choice functions, in Seiki AKAMA, editor, *Logic, Language and Computation*, volume 5 of *Applied Logic Series*, pp. 61–91, Springer.

J. Roger HINDLEY and Jonathan P. SELDIN (2008), *Lambda-Calculus and Combinators, an Introduction*, Cambridge University Press.

Jaako HINTIKKA (1962), *Knowledge and Belief. An Introduction to the Logic of the Two Notions*, Cornell University Press.

Kryštof HODER and Andrei VORONKOV (2011), Sine qua non for large theory reasoning, in Nikolaj BJØRNER and Viorica SOFRONIE-STOKKERMANS, editors, *Automated Deduction – CADE-23*, volume 6803 of *Lecture Notes in Computer Science*, pp. 299–314, Springer.

Joe HURD (2002), An LCF-style interface between HOL and first-order logic, in Andrei VORONKOV, editor, *Automated Deduction – CADE-18*, volume 2392 of *Lecture Notes in Computer Science*, pp. 134–138, Springer.

Bjørn JESPERSEN and Giuseppe PRIMIERO (2013), Alleged assassins: realist and constructivist semantics for modal modification, in Guram BEZHANISHVILI, Sebastian LÖBNER, Vincenzo MARRA, and Frank RICHTER, editors, *Logic, Language, and Computation. 9th International Tbilisi Symposium*, number 7758 in Lecture Notes in Computer Science, pp. 94–114, Springer.

Hans KAMP and Barbara PARTEE (1995), Prototype theory and compositionality, *Cognition*, 57(2):129–91.

Hans KAMP and Uwe REYLE (1993), *From Discourse to Logic*, Kluwer, Dordrecht.

Manfred KERBER (1992), *On the Representation of Mathematical Concepts and their Translation into First Order Logic*, Ph.D. thesis, Universität Kaiserslautern.

Michael KOHLHASE and Karsten KONRAD (1998), Higher-Order Automated Theorem Proving for Natural Language Semantics, unpublished manuscript. Web. Sept. 17, 2015.
`http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.54.4186`.

Karsten KONRAD (2004), *Model Generation for Natural Language Interpretation and Analysis*, number 2953 in Lecture Notes in Artificial Intelligence, Springer.

Daniel LEIVANT (1994), Higher order logic, in *Handbook of Logic in Artificial Intelligence and Logic Programming (2)*, pp. 229–322.

David K. LEWIS (1968), Counterpart theory and quantifed modal logic, *Journal of Philosophy*, 65(5):113–126.

Bill MACCARTNEY (2009), *Natural Language Inference*, Ph.D. thesis, Stanford University.

William MCCUNE (2005–2010), Prover9 and Mace4, Web. Sept. 17, 2015.
`http://www.cs.unm.edu/~mccune/prover9/`.

Jia MENG and Lawrence C. PAULSON (2008), Translating higher-order clauses to first-order clauses, *Journal of Automated Reasoning*, 40(1):35–60.

Richard MONTAGUE (1970), English as a formal language, in Bruno VISENTI, editor, *Linguaggi nella Società e nella Tecnica*, pp. 189–224, Edizioni di Comunità, Milan.

Richard MONTAGUE (1973), The proper treatment of quantification in ordinary English, in K. J. J. HINTIKKA, J. M. E. MORAVCSIK, and P. SUPPES, editors, *Approaches to Natural Language*, pp. 221–242, Reidel, Dordrecht.

Ekaterina OVCHINNIKOVA (2012), *Integration of World Knowledge for Natural Language Understanding*, volume 3 of *Atlantis Thinking Machines*, Atlantis Press.

Barbara H. PARTEE (1995), Lexical semantics and compositionality, in Lila R. GLEITMAN and Mark LIBERMAN, editors, *An Invitation to Cognitive Science. Language*, volume 1, pp. 311–360, MIT Press.

Allan RAMSAY (1995), Theorem proving for intensional logic, *Journal of Automated Reasoning*, 14:237–255.

Nicholas RESCHER (2005), *Epistemic Logic: A Survey of the Logic of Knowledge*, University of Pittsburgh Press.

Stephan SCHULZ (2004), System Description: E 0.81, in David BASIN and Michaël RUSINOWITCH, editors, *Automated Reasoning. Second International Joint Conference, IJCAR 2004*, volume 3097 of *Lecture Notes in Computer Science*, pp. 223–228, Springer.

Ray TURNER (1987), A theory of properties, *Journal of Symbolic Logic*, 52(2):455–472.

Johan van Benthem and Kees Doets (1983), Higher-order logic, in *Handbook of philosophical logic*, pp. 275–329, Springer.

Christoph Weidenbach (2001), Combining superposition, sorts and splitting, in Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume II, chapter 27, pp. 1965–2013, Elsevier Science.

Dag Westerståhl (2011), Generalized quantifiers, in Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, summer 2011 edition.

Entropic evolution of lexical richness of homogeneous texts over time: A dynamic complexity perspective

Yanhui Zhang
The Chinese University of Hong Kong

ABSTRACT

This work concerns the evolving pattern of the lexical richness of the corpus text of China Government Work Report measured by entropy, based on a fundamental assumption that these texts are linguistically homogeneous. The corpus is interpreted and studied as a dynamic system, the components of which maintain spontaneous variations, adjustment, self-organizations, and adaptations to fit into the semantic, discourse, and sociolinguistic functions that the text is set to perform. Both the macroscopic structural trend and the microscopic fluctuations of the time series of the interested entropic process are meticulously investigated from the dynamic complexity theoretical perspective. Rigorous nonlinear regression analysis is provided throughout the study for empirical justifications to the theoretical postulations. An overall concave model with modulated fluctuations incorporated is proposed and statistically tested to represent the key quantitative findings. Possible extensions of the current study are discussed.

Keywords: dynamic complexity, lexical richness, entropy, homogenous texts, language modeling

1 INTRODUCTION

Corpus linguists and experts in related fields have shown increasing interest in homogeneous texts, largely because homogenization is often an effective and statistically trustful way to filter out the unnecessary or, even worse, the distorted information from the raw meta-corpus data, thus helping to uncover the principal linguistic variables as well as the governing laws that a researcher is keen to find. Study surrounding homogeneous texts can be undertaken from many perspectives, including homogeneity measurement, corpus selection, and applications in language acquisition and sociolinguistic analysis. For instance, the cross-corpora studies of Kilgarriff (2001), Kilgarriff and Grefenstette (2003), and Denoual (2005) relied heavily upon the notion of homogeneity. Kornai *et al.* (2006) focused on texts' homogeneity characterized by their stylistic features, particularly those discernable through author tags. Crossley and McNamara (2011) used word-based indices such as hypernymy and stem overlap to test the intergroup homogeneities among L2 English learners and cross-group heterogeneities between L2 and L1 writers so as to facilitate the understanding of the development of L2 writing. Sahlgren and Karlgren (2005) confined homogeneity to the extent of topical dispersion with empirical applications. The primary interest of the current study is to understand how the complexity of a given set of homogeneous texts progresses over time. For this purpose, the corpus is treated as an interacting, adaptive, and constantly evolving system, the evolution of which is regulated by the internal linguistic laws as well as external sociocultural conditions at large.

Lexical richness, a primary indicator of verbal variation and sophistication and hence the degree of complexity profiled by an interested text, is a particularly useful tool for quantitative and computational linguistics, the application of which can be found in Smith and Kelly (2002) for author attribution and in Johansson (2008) for language proficiency assessment. Existing literature on lexical richness is mostly concerned with the impact of spatial factors, such as how lexical richness is influenced by different writing styles or how lexical richness will vary as text length increases. This includes the above-mentioned references in this paragraph and the classic work of Shannon (1951), where maximum entropy of English was analyzed from an information science perspective, as well as the more recent works of Brown *et al.* (1992) and Genzel and Charniak (2002) with a similar focus. For all such examples, the data used and the core questions under investigation are cross-sectional, i.e., they are concerned with linguistic features at a fixed time, even though the dimension and contributing factors can be complicated.

The current study is fundamentally different in that it focuses on the evolving structure of the lexical richness over a large span of time. In other words, it is dealing with large-scale longitude data instead of static data at a fixed time. The study investigates the lexical richness properties of a sequence of homogeneous texts, namely, the texts of China Government Work Report (CGWR) spanning from 1954 to 2011. The entropies of these texts are calculated and treated as a time series data. Under the framework of the dynamical complexity theory, the study analyzes and accurately depicts how the entropy of the CGWR texts progresses in a time span of over fifty years. Adequate probes into the data and the regression results allow us to trust on a concave and upper bounded exponential model to describe the observed entropy evolving process. Further diagnostication of the model approbates the differentiation of the whole process into two phases, namely, an initial phase where the entropy grows sharply with vehement fluctuations and a maturing phase where the process approaches a stationary baseline with small, minuscule fluctuations, where the fluctuations can be modeled by wavelet trigonometric functions. Interpolation of the initial concave exponential growth and the modulated fluctuations at the maturing phase yields a unified model that captures both the long-term trend and the local variations.

The rest of the paper unfolds as follows. Section 2 explains the CGWR corpus used for the study, followed by a preliminary analysis of the raw entropy data of the corpus. Section 3 briefly describes the dynamic complexity theory and its applications in related areas, on the basis of which postulations are drawn regarding the evolving pattern of the entropic process under review. Section 4 presents a mathematical model for capturing the global structure of the time series of the entropy data, followed by a rigorous assessment of the validity of the model. Section 5 is set out to improve the model's accuracy and predictive power by incorporating the local microscopic fluctuations of the process. Concluding remarks and possible future directions are discussed in Section 6.

2 CORPUS, MEASUREMENT, AND DESCRIPTIVE STATISTICS

2.1 *Corpus of CGWR*

The corpus used for the study consists of the CGWR written texts archived from 1954, when the first CGWR was published, to 2011, excluding the years that the CGWR was not issued: 1961–1963, 1965–1974, and 1976–1977. Each text contains on average 22,373 Chinese characters with a standard deviation of 8256.5, making the size of the corpus approximately 962,000 characters in total. The archives of the CGWR corpus are publicly accessible at the webpage of the central government of China *www.gov.cn*. The CGWR, as one of the most important public documents in China, is drafted in accordance to a stable and formatted style, covering various major aspects of the sociocultural, political, and economic life at national level, as well as the events and projects of significance of the corresponding year.

While the sociopolitical importance of the CGWR texts is self-evident, it is their linguistically homogeneity feature that most concerns the current study. Although there exist studies such as Gries (2006) suggesting using complex techniques to quantify homogeneity, the notion of homogeneity in corpus linguistics appears rather wide and informal, as felt by Kilgarriff (2001), for instance. As to the CGWR texts in the current study, they are topically homogeneous from year to year although the emphasis may vary. They are drafted by the same institutional author whose writing style seems to be even more consistent than texts by individual authors. Moreover, the production of CGWR is periodic and subject to a strict scrutiny and modification process set by both linguistic norms and political operations.

2.2 *Entropy measure for lexical richness*

Lexical richness refers to the size of the vocabulary that is employed in language generation and how diversely the words are used. Intuitively, it reflects the degree of variations and sophistications of a spoken or written text, the production of which must of course adhere to the constraints and rules imposed by the language being used. While lexical richness is something that can be either clearly or vaguely perceived in daily conversations, assigning a numeric value to it becomes indispensable when scientific research of corpus linguistics is being conducted on a massive scale. The numeric measure adopted in the current study to quantify the lexical richness of the CGWR texts is entropy, the concept of which originates from physical sciences, particularly thermodynamics.

Consider a Chinese corpus text, denoted as T, which has n different characters indexed with 1, 2, ..., n. Assume that the relative frequencies of each of the n characters appearing in the corpus are p_1, p_2, ..., p_n, then the entropy of the Chinese text is defined as

$$\mathit{Entropy}(T) = -\sum_{i=1}^{n} p_i \ln(p_i).$$

Originally introduced in thermodynamics for quantifying the unpredictability of the microscopic state of a physical system at any given time, entropy has now become a widely accepted concept and a tested measure of uncertainty and/or complexity in many disciplines and interdisciplinary fields such as communication science, ecology, biology, and cosmology, to name a few. As useful as it is, what entropy really measures can be dependent on the context of use and field knowledge of specific disciplines. In particular, it could be naïve to treat the entropy in classical thermodynamics as equivalent to the Shannon entropy, despite that they take the same form in calculation. An insightful ontological discussion on entropy can be found in Wicken (1987), for instance. On the other hand, when used for quantifying lexical richness as in the current study, entropy should be best understood as a measure of the degree of complexity that the original system, usually composed of finite components and limited number of laws governing the interactions between the components, has developed as of today. For a fixed time horizon, what is emphasized here is the compositional complexity of the linguistic construct of a text (Jarvis 2013).

It is a simple calculation, using the above formula, to show that the maximum possible value of entropy for T is achieved when all the characters in it are different from one another, in which case $\mathit{Entropy}(T) = \ln(n)$, where n is both the total number of characters (tokens) and the number of unique characters (types) appearing in the text. Nevertheless, the entropy of any meaningful text is in reality far below this number because, first, the total number of unique Chinese characters (or the total number of types of any language in general) is capped; and second, the distribution of all the unique characters (or the types of any language) is far from, not even close to, uniform distribution. As a matter of fact, the second rationale of the above partly echoes the well-known Zipf's law. Take the CGWR of 1954 as example, Table 1 provides a summary of key statistics relevant to the current study. And Figure 1 pro-

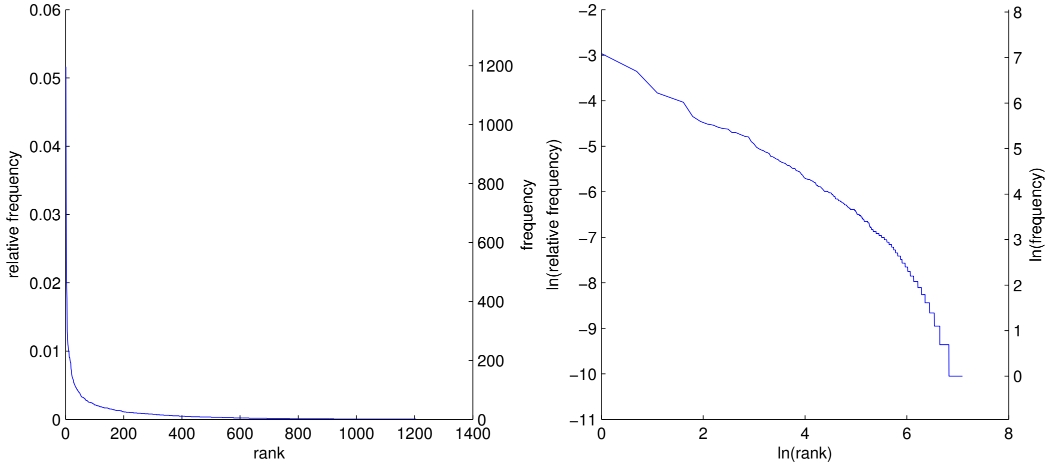


Figure 1: Frequency distribution plots of CGWR 1954 text

vides the corresponding frequency plots, where the left plot is in the original scale, and the right plot is scaled by the natural logarithm.

Table 1: Descriptive statistics of the CGWR 1954 text

Year	Total unique characters	Total characters	TTR	Entropy	Maximum entropy
1954	1205	23168	0.052	5.8601	10.0505

3 THEORETICAL FRAMEWORK AND RELATED RESEARCH

3.1 *Overview of dynamic complexity theory*

The core theoretical foundation that forms the basis for the assumptions of the current paper, and according to which the statistical models are constructed, is the theory of the dynamic complexity system. The theory, despite its diverse origins and applied fields, is formulated and commonly accepted nowadays insofar as it *corrects* the tendency in classical approaches in physical sciences to explain both natural and human phenomena with over-simplified assumptions and static mechanisms. Given its multidisciplinary and interdisciplinary nature, it is not easy to portray a full genealogy of dynamic complexity (some antecedents of complexity theories from linguists' perspective can be found in Larsen-Freeman and Cameron 2008, pp. 2–4). Early mathematical usage of complexity using the concept of entropy is usually traced back to classic thermodynamics (Bailyn 1994), the focus concern of which is how heat is transferred in a physical system and how the system evolves in the irreversible time direction. Dynamic complexity is, in a sense, a general postulate of the second law of thermodynamics in broader disciplines beyond physics and chemistry.

It is important to keep in mind that the complexity system contextualized in contemporary scholarly research is far more "complicated" and multifaceted than its counterpart in thermodynamics. Among others, one notable difference is that traditional thermodynamics only deals with an isolated physical system, allowing no matter or energy exchange across the boundaries. Hence, the law governing the entropy process therein, as complex as it can be, is deterministic. Fundamentally different from classic sciences, the dynamic complexity theory used in this study views any examined entity as a complex and constantly evolving system, the members or components of which are interacting with each other, each evolving as a sub-system under the constraints imposed by the system as a whole. Exchange of matter, energy, and information is allowed not only among the interacting make-ups, but also between the system and the external environment in which the system is sustained. Almost as a consequence, it allows for self-organization, chaos behavior, nonlinear progression, and phase changes (Larsen-Freeman and Cameron 2008).

3.2 *Application in related research*

Nowadays, dynamic complexity theory has proven a useful framework for many applied fields in physical and social sciences. Direct or indirect introduction of dynamic complexity into studies of linguistic phenomena has led to fruitful results on a number of frontiers, particularly in the past two decades. For example, a dynamic language development approach was taken by Verspoor and Behrens (2011) to explain the role of frequency in L1 learning and the role of L1 in L2 learning. Spivey (2007) asserted the continuity of mind, emphasizing the dynamic and complex characteristic of human's cognitive, hence linguistic function. Meara (2006) adopted a similar approach for modeling vocabulary learning. A thorough treatment of linguistic complexity theory is presented in Larsen-Freeman and Cameron (2008), where the core rationales and properties defining “complexity systems” in language study are meticulously laid out. Many studies, such as Blevins (2004), Croft (2008), and Lee and Schumann (2003), fall within the framework of evolutionary linguistics, which partly overlaps the idea of dynamic complexity theory, particularly when the self-adaptive nature of languages is underscored. A similar approach was taken by Wang (1979) in accommodating the diffusions and randomness observed in language changes. Dynamic complexity is also presented in the competition model developed by MacWhinney (2007) in accounting for the spectrum of interrelated phenomena arising from FLA and SLAs. Useful as they are, the applications of the dynamic complexity theory in most of the existing studies are lacking a unified measure, and the analysis to date has been mostly qualitative in nature. Our statistical modeling, in part, exemplifies an attempt to bridge this gap in the focused area of corpus linguistics.

3.3 *Pertinence to CGWR*

According to Larsen-Freeman and Cameron (2008), a complex system is “a system with different types of elements, usually in large numbers, which connect and interact in different and changing ways” (p. 26). While others such as Verspoor *et al.* (2011) have summarized in different ways, virtually all the theorists agree that dynamicity and spontaneous changes between both interconnected elements as well as the system as a whole are the central property for a system to be complex. For the CGWR to be characterized as such, the constituent agents, from a complex system perspective, are the Chinese characters, words, phrases, idioms, and proper nouns commonly related to the sociocultural, political, and economic life of contemporary China. Not only are these components completely interconnected and interacting with each other spontaneously, but the discourse structure and rhetoric strategies pertaining to them are also constantly changing to fit the linguistic functions that the CGWR text is supposed to perform. When an entropic metric is imposed macroscopically, the system is unsurprisingly manifested as a self-adaptive process, evolving from simple primitive forms to more complicated ones under regulations of both

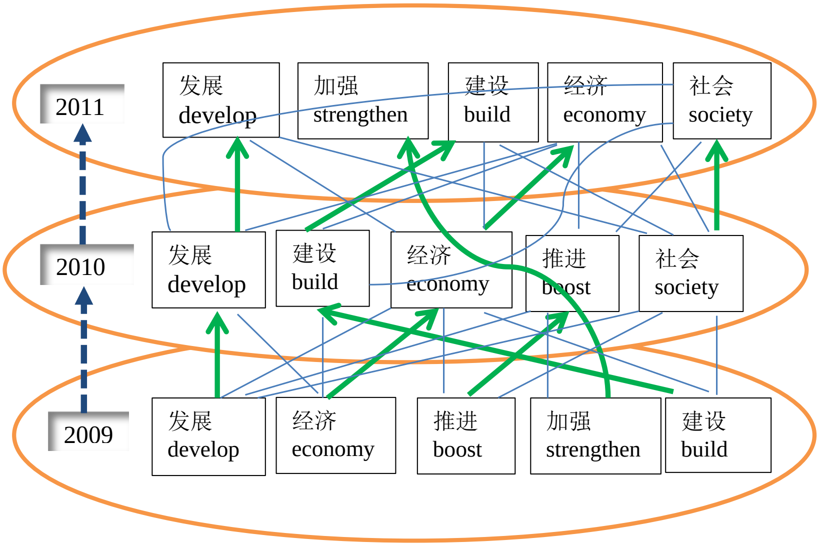


Figure 2: Most frequent content-words in the CGWR texts of 2009–11

formal linguistic rules and peripheral sociolinguistic norms of the society.

Figure 2 provides a diagram of the changes in the content-words appearing most frequently in the CGWR texts from the years 2009, 2010, and 2011, which aptly reveals the dynamic quality of the CGWR. At least three major factors contributing to the relative changes in the ranking and frequency of these content-words can be identified as follows. First is the topic continuation of CGWR over time, represented by the thick solid arrows (in green) in the diagram. For example, “to develop” or “development” played a central topical role in the CGWR in the three years under analysis; it was consistently the most frequently occurring content word across the CGWR texts during all three years (117 for 2009; 123 for 2010; 139 for 2011). Other notable topical words include “economy”, “to build”, and “to strengthen”, the relative usage of which saw more fluctuations. Second is the dynamics of lexical networks over time, denoted by the thin curve segments (in light steel blue) in the diagram, where an edge in the network can be defined by synonyms such as *tui1jin4* (to boost) and *fa1zhan3* (to develop), for instance; or a syntactic dependency as in the concurrence of *fa1zhan3* (to develop) and *jing1ji4* (economy), for instance. The third factor figuring in the dynamic quality of the

CGWR consists of the complexity explicated by the social, cultural, political, and economic contexts in which the CGWRs were drafted. This type of complexity, conceptualized by the ellipses as well as the thick dashed arrows (in dark steel blue) between such ellipses in the diagram, reflects the co-adaptive nature of the CGWR, where it allows for the exchange of energy and matter across the boundaries and draws on resources and influences from the external sociocultural environment in general. As such, a full understanding of the linguistic dynamism of the CGWR texts is not probable without reference to the parallel social, cultural, political, and economic realities of the society.

Of course Figure 2 is far from complete in depicting the infinite microscopic complexities belonging to the system under study. It only provides a glimpse, from a rather limited angle, of the vast lexical dynamics present in the CGWR texts from year to year. Many subtle changes caused by lexical inertia or a variety of cohesions are not easy to describe accurately, neither can the emergence of new words driven by technology advancement or socioeconomic shifts, for instance, be fully accounted for. Nevertheless, despite the lack of a complete microscopic description, the dynamic nature of the CGWR texts is sufficiently evident from this illustrative diagram. After all, the macro evolving pattern instead of the micro and local cause is the focus concern of the current investigation. Moreover, the goal of a dynamic approach, according to Verspoor *et al.* (2011), "is not to list possible causes for change and development but to describe the process of change and development itself by means of tracing the iterative change over time". Table 2 identifies the key properties of the CGWR serving to define its dynamic complexity nature. The items in the Field column of the table were pointed out by Larsen-Freeman and Cameron (2008) as the defining features of a system being complex. The second and third columns of the table are adapted from the same reference (p. 37).

4 GLOBAL ENTROPIC MODEL FOR CGWR TEXTS

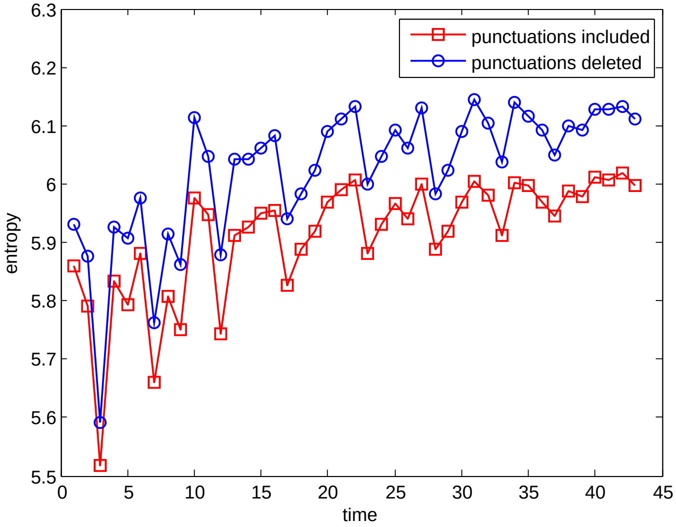
To properly envision a mathematical model that appeals to the dynamic nature of CGWR explained in the previous section and simultaneously captures its general entropic evolution pattern, it is reason-

Table 2: Defining features of CGWR and other complex systems

Field	Ecology	Classroom language learning	CGWR
Agent	individual animals	students, teachers, languages	characters, words, proper nouns
Heterogeneity	eating, nesting, breeding, habits	abilities, personalities, learning demands	meaning, lexical relationships
Organization	schools, herds, food chains	class, groups, curricula, grammars	content vs. function, part of speech, thematic group
Adaption	hunting, mating, security	imitation, memorizing, classroom behaviors	derivation, metaphor, situational context
Dynamics	predator-prey interactions, competition	classroom discourse, tasks, participation patterns	rhetorical force, styles, sociocultural influence
Emergent behavior	extinction, niches	language learning, class/group behavior, linguae francae	internet language, word fashion, popularity

able to start with a qualitative exploration of the empirically observed data. Figure 3 presents the scatterplots of the calculated entropies pertaining to the CGWR texts, where time denotes the number of years since 1953, skipping those years in which the CGWR was not issued, as pointed out in section two (the same definition applies to all the subsequent models and plots). The upper plot in Figure 3 corresponds to the data set with all punctuation deleted, and the lower plot to the data set with all punctuation included. These two series show very similar tendencies, but the entropy values for the data containing all punctuation are systematically lower than those with all punctuation deleted. The reason for such a difference is that punctuation constitutes extra linguistic constraints imposed on the text; and according to the dynamic complexity theory, the more imposed constraints, the lower

Figure 3: Scatterplot of the entropic processes



diversity of a system, with other conditions fixed. Whether punctuation should be included or not depends on the purpose of study. There exist examples where punctuation spaces are ignored (Shannon 1951, for instance) and also examples where they are included (Brown *et al.* 1992). For the subsequent analysis, all models are constructed with punctuation included, but they are equally valid for the scenario with punctuation deleted.

4.1 *Some observations*

It is a palpable observation that CGWR, as a dynamic complex system, is generally increasing in entropy. The ascending trend of the entropic process is first a manifestation of the increasing complexity of CGWR in terms of lexical choice, syntactic structuring, and discourse planning. It reflects the many and changing ways that all such constituents can interact, mutate, and concatenate with each other. To be able to appreciate this overall pattern, it helps to realize that a third-party reader will more likely to encounter new words, advanced semantic constructs, sophisticated cohesions, unprepared concepts, etc. when reading the CGWR texts in chronological order. On the other hand, CGWR is inseparably connected into the social and societal dynamics it purports to describe. This sociocultural-ecological perspective of languages (see Steffensen and Fill 2014; also Larsen-Freeman and

Cameron 2008) allows us to view the CGWR as a linguistic vessel of the society's events and histories. Ideally, the entropic process of the CGWR text shall behave in the same way as the entropic process of the societal focus it depicts, although it is quite unlikely, in reality, for such dual processes to be exactly parallel to each other. Consequently, as the complexity of human society increases (technologically, culturally, and economically), so does that of the associated linguistic agents such as the CGWR under study.

On the other hand, because the interacting linguistic components such as characters or punctuation must maintain certain lexemic, etymological and grammatical structures so as to sustain the linguistic functions of the system, the rate of increase of entropy of a homogeneous text with a roughly constant size will eventually decline, constrained by the linguistic and sociocultural conditions. Analogous arguments apply to the dual process of human society. Although interactions between parts, self-organization, randomness, nonlinear behaviors, even chaos and bifurcations are allowed in human organization, the level of possible complexities must be capped due to the constraints of, for instance, laws, cultural norms, limited capacity of production, and ethnic bonds. These conditions and constraints are necessary to conserve the defining properties of the system and prevent it from malfunctioning.

Lastly, one should expect fluctuations in the entropic process of the CGWR texts. This is different from the classic statement of the second law of thermodynamics, in which the entropy is asserted to be monotonically increasing. The difference is that a government publication is not an isolated system. Instead, it needs to accommodate the addition or deletion of lexicons, and must also allow for an inflow of foreign discourse styles, among other elements. Furthermore, such a text is subject to artificial modifications in terms of topic, theme, or size of text, in reaction to the changes in the human society system it endeavors to depict. Additionally, the fluctuations of the entropy process of a homogeneous text should be vehement in the initial stage and moderate in the maturing stage. The rationale for this, from a self-organization perspective, is that the initial stage of a system retains much less structural inertia than the maturing stage. Thus, random and nonlinear mechanics can cause dramatic changes to an emerging system with much less cost.

To summarize, the overall trend of the entropic process of the homogeneous CGWR texts is an upper bounded increasing function in time, where the trend undergoes a fast increasing initial phase before flattens to a saturated phase in the long term. In addition, fluctuations are accompanied throughout the whole process, where the magnitude of the fluctuation is large for the initial phase and small for the saturated phase.

4.2 *A basic model for the principal trend*

For quantitative modeling purpose, the study embraces a mathematical function having the following characteristics: 1) it increases rapidly in the beginning, plateauing as time goes on; 2) it is upper bounded and eventually flattens to a horizontal line, which is the upper limit of the expected entropy for the given length of the text. This leads to the following choice of equation (1), a generalized exponential type function with such postulated growth patterns:

$$E = b_1 - b_2 e^{-b_3 t}, \tag{1}$$

where t denotes time (measured in years) since the beginning of the practice of CGWR, and e is the exponential function. The same notation and definition apply to following equations and discussions. The choice of model (1) is not for the convenience of data analysis, although the exponential function, the second term of (1), is indeed a built-in class in many statistical packages, such as SPSS. It is selected because many natural phenomena, including those in linguistic processes, have been shown to develop in that way. For instance, Szmrecsanyi (2005) showed how the percentage of persistent pairs in a text, as a function of the textual distance of the pair, is decreasing exponentially. Learning effectiveness of repetition priming was reported to decay exponentially as a function of the length of the lag time (McKone 1995). Beeferman *et al.* (1997) provided an empirical study on why a model of exponential type can be used to describe the attractive and repulsive distances between word pairs with high mutual information. Despite these almost ubiquitous exponential phenomena being observed, a potential criticism might still be raised that all such examples are modeling a decaying process rather than an increasing one. But one should note that the usual decay model taking, for example, the form $y = ae^{-bt}$, is actually a special case of (1) when the signs of the parameters are not restricted. There is a bound parameter for the usual decay model also, in the sense that zero is its lower bound.

The procedure to find the best estimates of the parameters appearing in model (1) as well as a procedure for model evaluation, under the normality assumption of errors, can be facilitated by statistical packages such as Matlab or SPSS. Precautions still need to be taken, though, in terms of choosing reasonable initial guesses of the target parameters and avoiding common pitfalls associated with nonlinear regression, e.g., over-fitting. The least square optimization procedure yields the following estimates for the model defined by equation (1):

$$\begin{aligned} b_1 &= 6.0222 \\ b_2 &= 0.3089 \\ b_3 &= 0.0580 \end{aligned}$$

The corresponding standard errors for the parameter estimates are 0.0558, 0.0483, and 0.0301. The R-squared and adjusted R-squared statistics for the model are 0.5406 and 0.5177, respectively. Although the R-squared value does not seem impressively high, one should keep in mind that the validity of a nonlinear model is not solely, and not even largely, determined by the magnitude of the R-squared value when the general trend of a process is the main concern of a study. For a more rigorous explanation of why the R-squared value should not be a main concern in trend analysis, one can refer to Wittink (1988). On the other hand, the R-squared value of the basic model (1) can indeed be improved, as discussed in the next section. The t-statistics for parameters are 107.8670, 6.3982, and 1.9262, with corresponding p-values of 0.0000, 0.0000, and 0.0305 (accurate to four decimal places), respectively. Clearly each t-statistic is large enough and each p-value is small enough, which strongly justifies the statistical significance of each individual parameter in model (1). The calculated F-statistics for the model is 24.1258, with the corresponding p-value of 1.1870×10^{-7}. The overall explaining power of the model is strong. Figure 4 plots the fitted curve of the global model, together with a 95% confidence band of the regression.

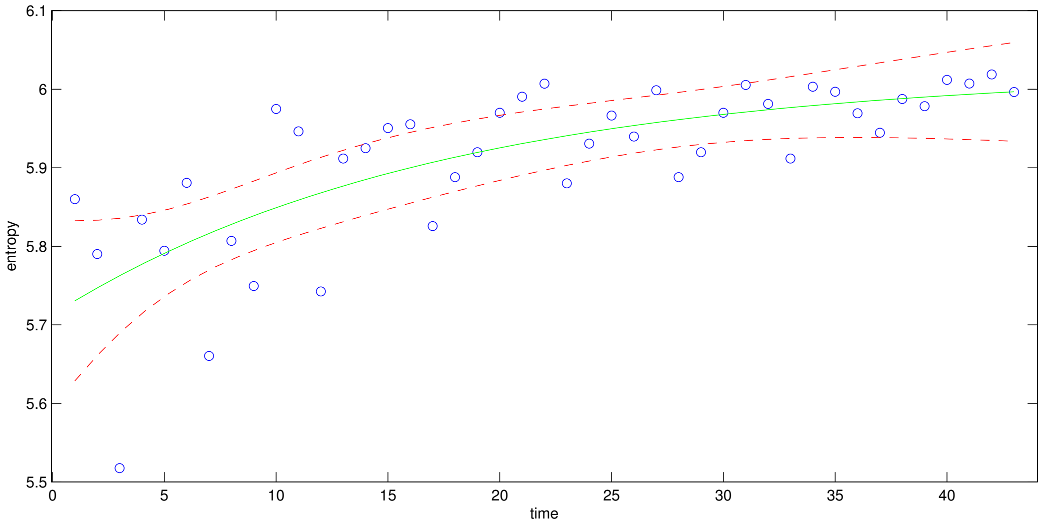


Figure 4: Plot of the original data (the circular points), basic global model (the connected curve), and 95% confidence band of the regression (the region between the dashed curves)

4.3 *Model the local fluctuations*

The backbone structure of the dynamic evolution of the entropy process is implied by the concave exponential model defined by equation (1), yielding not only a quick growth feature in the beginning of the process, but also a quick plateau effect when time is large. This said, the model does not capture the microscopic structure of the process, which exhibits large or small fluctuations at all times. There are standard statistical methods that might help to improve the model accuracy, such as smoothing and autoregression. But a relatively simple and more direct approach is to introduce the wavelike functions to the model, namely, trigonometric sine or cosine functions.

Notice the fluctuation of entropies is initially more vehement and becomes moderate later on as the process approaches steady state. It is therefore plausible to separately analyze the process in two stages: an initial quick growth stage where the process is more volatile, and the steady stage where the growth momentum is mild and the fluctuation is moderate. A clue to this can be obtained by an expository check of the scatterplot of the entropy, where the 12th data point appears to be the borderline after which the series becomes relatively stationary. To validate statistically, one can appeal to the unit root test, a standard procedure in time series analysis for testing whether a given series is stationary or not at a prescribed level of confidence. The procedure applied to the 31 observations, i.e., the suggested steady stage of the original entropic time series, rejects the null hypothesis that the series under testing has a unit root, or equivalently affirms the hypothesized stationary nature of it, and does so at a 90% confidence level. To be specific, the augmented Dickey-Fuller statistic is −2.7603. The critical values of the test are −2.6210, −2.9640, and −3.6702 at, respectively, the 90%, 95%, and 99% confidence levels.

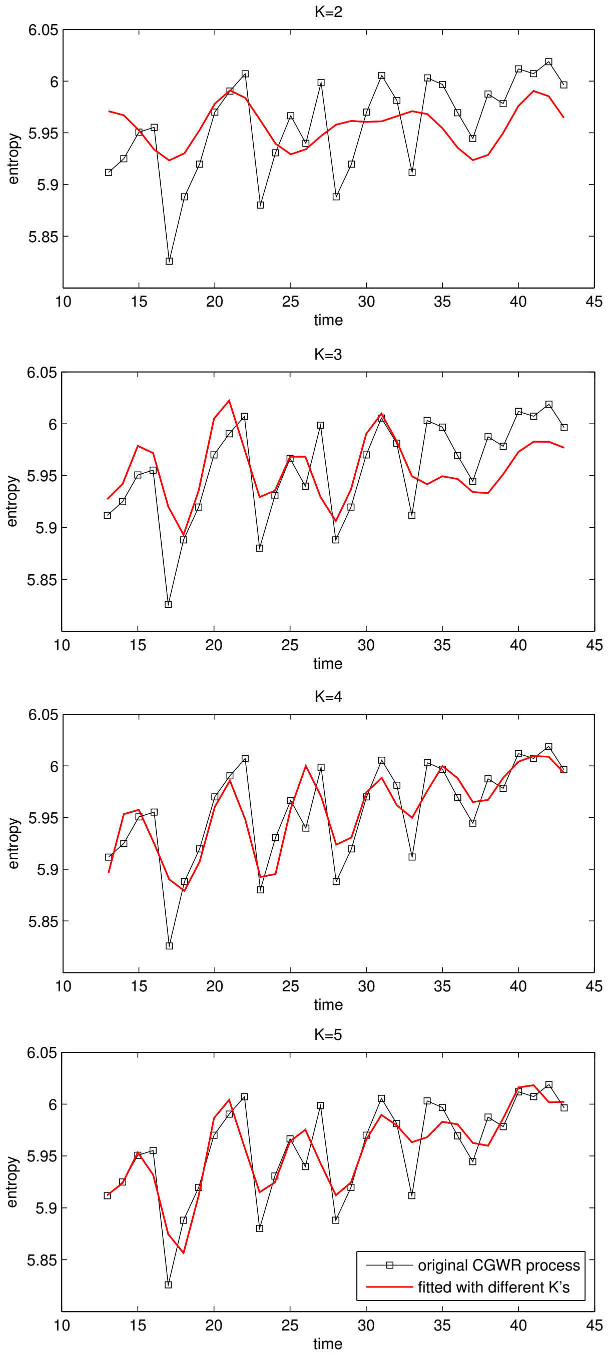
Now we turn to the modeling of the steady state with a cutting off point set at the 12th point. To model this truncated series of data with the fluctuation characteristics being the core concern, the following trigonometric functions are chosen:

$$E = k_0 + \sum_{j=1}^{\mathrm{K}} k_1(j)\sin(k_2(j)t + k_3(j)). \tag{2}$$

When K is specified, the parameters can be determined using the same regressing procedure carried out for the model defined by equation (1), and the regression evaluation can be performed accordingly. The only new issue that may complicate the procedure is the choice of K, which defines how many trigonometric functions are to be used in the model. With homogeneity of the data and validity of the model (2) in mind, one can apply an iterative scheme to find such an optimal K numerically. For the current analysis, the following rules of thumb were followed in selecting the optimal K, namely, i) the increase in adjusted R-squared divided by the increase in R-squared value is approaching maximum; ii) the majority, if not all, estimates are significant enough, judging by the corresponding t-statistics or p-values; iii) the overall F-statistics for the model is significant enough. By these rules of thumb, K = 4 is found to be optimal for the model under review. Figure 5 presents the comparison plots for cases K = 2, 3, 4, 5. Table 3 summarizes the key regression statistics, where K = 4 is observed as the choice of how many sine functions to include for best fitting the data.

One comment to add is that there appears to be a relatively large gap between the R-squared value and the adjusted R-squared value.

Figure 5:
Plot of the steady state series and the fitted curves using trigonometric functions with K = 2, 3, 4, 5



	K = 2	K = 3	K = 4	K = 5
R-squared	0.1801	0.4197	0.6761	0.7131
Adjusted R-squared	0.0248	0.1710	0.4602	0.4262
F-statistics	0.9155	1.7682	3.3054	2.6515
p-value for F-test	0.5002	0.1325	0.0099	0.0309

Table 3:
Regression statistics for the steady state series with different K

This is mainly a consequence of the small size of the data set. When the sample size is large enough compared to the number of independent variables, the adjusted R-squared value should be virtually the same as the R-squared value itself. This also implies that the model will work better as the CGWR corpus increases in size.

5 IMPROVED GLOBAL MODEL

The improvement of the global model can be achieved by a consolidation of the overall concave exponential structure and the trigonometric microscopic structure of fluctuations. In other words, the exponential component and the trigonometric component jointly depict the evolution of entropy with high resolution at local and global levels. Specifically, the following model is proposed for this purpose:

$$E = b_1 - b_2 e^{-b_3 t} + b_4 \sin(b_5 t + b_6) e^{-b_7 t}. \qquad (3)$$

The product term of the exponential factor and the trigonometric factor corresponds to the interactions between the general trend and local fluctuations. The magnitude of the wavelets yielded by the product term is high when t is small, and low when t is large, making the term a suitable choice to describe the fluctuations observed in the CGWR process. The parameter estimation procedure is same as that applied to model (1). Actually the values of the estimated parameters for model (1) can be used as part of the initial guesses for the parameter vector for model (3). Going through the nonlinear regression procedure leads to the following parameter estimations:

$$\begin{array}{ll} b_1 = 6.0169 & b_2 = 0.3105 \\ b_3 = 0.0620 & b_4 = 0.1721 \\ b_5 = 1.2703 & b_6 = 0.9612 \\ b_7 = 0.0740 & \end{array}$$

The model is statistically significant, with a sound explaining power, as shown in all critical aspects of observed statistics under scrutiny via various standard tests. The t-statistics for all the parameters b_1—b_7 are sufficiently large and the p-values for all the parameters $b_1 - b_7$ are sufficiently small, where the observed smallest t-statistic is 2.4950 (for the parameter b_3), corresponding to a p-value of 0.0086. The significance of each parameter can also be assessed by how far away the confidence interval of the estimate is from zero, at the prescribed confidence level. Computation shows that the 90% confidence intervals for all the parameter estimates are far enough from zero. For instance, the ratio between the estimate of b_3 and the corresponding half confidence interval is about 1.4797; and it is the lowest one among the seven such ratios. The R-squared and adjusted R-squared values are 0.7432 and 0.7004 respectively, both at acceptable levels for such a highly nonlinear model with frequent fluctuations. The overall validity of the model is particularly shown in the significance of the F-statistics, which is 17.8450, and the corresponding p-value, which is 1.3692×10^{-9}. In addition to the significance of each individual parameter $b_1 - b_7$, none of the paired correlations between the estimated parameters is higher than 0.8 in absolute value except for parameters b_3 and b_1, the correlation between which is about −0.93, implying that the model basically does not have the problem of parameter redundancy and over-fitting. The full correlation matrix of the estimated parameters for model (3) is provided in Table 4.

Table 4: The correlation matrix of the estimated parameters in the global model (3)

	b_1	b_2	b_3	b_4	b_5	b_6	b_7
b_1	1						
b_2	0.4527	1					
b_3	−0.9285	−0.1677	1				
b_4	−0.0462	−0.0229	0.0443	1			
b_5	−0.1310	0.1510	0.1864	−0.1057	1		
b_6	0.1885	−0.1980	−0.2620	0.1398	−0.7675	1	
b_7	−0.0576	−0.0212	0.0551	0.7454	−0.0727	0.0945	1

It can be verified, however, that one or more of the above conclusions will be violated or weakened when one or more trigonometric terms are added, which shows that the model in the current formulation is optimal in terms of how many corrective terms need to be in-

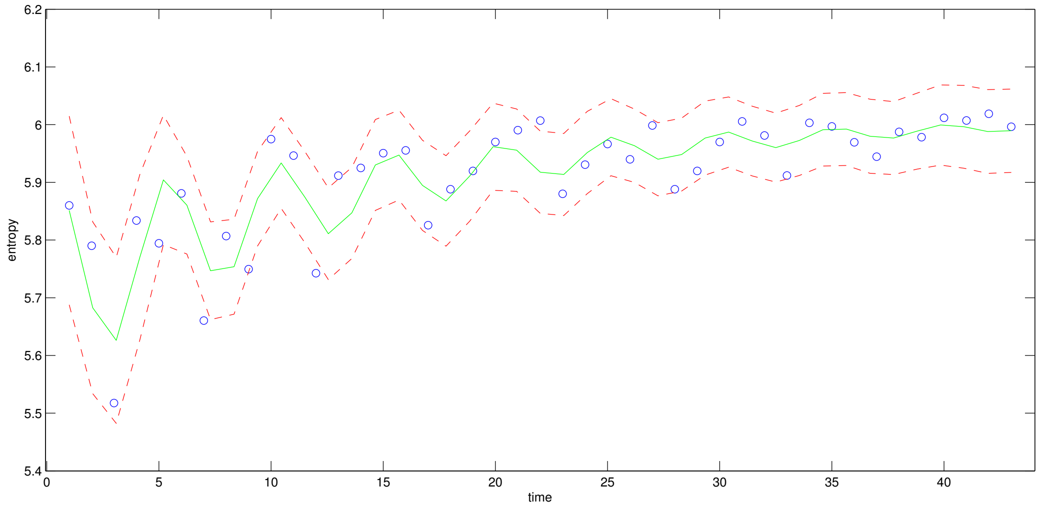


Figure 6: Plot of the original data (the circular points), improved model (the connected curve), and 95% confidence band of the regression (the region between the dashed curves)

corporated, given the current component functions and format of the model. For example, adding another trigonometric term can slightly increase the R-squared value to 0.7484, but the adjusted R-squared value will drop to 0.6214. Figure 6 plots the fitted curve and the corresponding 95% confidence band of the regression.

Although the above internal regression procedure appears to favor model (3), it is a reasonable concern that it does not overfit the data, especially because the available CGWR texts are relatively scant. The key issue here is whether adding more parameters into model (1) is a worthy effort when extrapolative prediction is also taken into consideration. Since we are mainly concerned with the evolution pattern of the CGWR text in the irreversible time direction, the appropriate numeric overfitting test to apply will be the out-of-sample prediction test, i.e., assessing which model can better forecast the future movement of the entropic process of the CGWR based on the past information. Many overfitting statistics have been developed and used for time series model selection. Here, I choose the three most widely used statistics, namely, mean squared error of out-of-sample prediction (MSE), mean absolute error of prediction (MAE), and mean absolute percentage error of prediction (MAPE) to compare the model (1) and (3). In addition, I present two more statistics for comparison: one is prediction error variance (PEV), measuring how consistent the errors are; and the other is the Theil statistic (Theil) measuring how relatively effective the model is compared to a naïve model, where the future value is simply predicted as the current value. For detailed discussion of these statistics as well as their relevance in assessing the overfitting of time series modeling, one can, for instance, refer to Bisgaard and Kulahci (2004) and Fildes (1992).

To carry out the out-of-sample cross validation, one needs to decide on a cutting point on the time direction. Thereafter the sequential data are used as the pseudo future cases against which the predicted values are compared. While the choice of the size of this test set is not completely rigid, this study follows the fourth quarter holdout rule, i.e., the rounding point of the 25% of the time series from the end as the starting point of the out-of-sample prediction test, which has been agreed upon by most of the theorists and practices for time series modeling (Hastie *et al.* 2009). Table 5 provides the 1-step-ahead and 3-step-ahead forecast statistics of the model (1) and model (3), where it is evident that the one-year forward movement of the entropic process is consistently more predictable with model (3) than with model (1) under all the chosen testing criteria.

Table 5: Out-of-sample test statistics for model (1) and (3)

		MSE	MAE	MAPE	PEV (10^{-5})	Theil
1-step-ahead forecast	model (1)	0.0009	0.0232	0.0039	0.9390	0.5255
	model (3)	0.0007	0.0222	0.0037	0.5535	0.4169
3-step-ahead forecast	model (1)	0.0004	0.017	0.0028	0.4450	0.7879
	model (3)	0.0004	0.0182	0.0030	0.3086	0.7778

It is worth noting that the Theil statistics for both models are at an acceptable level, affirming the usefulness of both the models, regardless of the difference in the out-of-sample prediction tests. On the other hand, the improvement in prediction accuracy of the model (3) does not seem to compensate for its increased complicatedness, when judged by the multi-step-ahead forecast statistics. To interpret these statistics, it is worthwhile to refresh the idea that "overfitting is not an absolute but involves a comparison” (Hawkins 2004). Similar precautions have been expressed by Bisgaard and Kulahci (2004) – that numerical and statistical tests of overfitting should not be applied mechanically without reference to the research contexts and purposes. Because the CGWR is viewed in the current study as a dynamic complexity system which intrinsically allows for fluctuations, model (3) appears a plausible choice when a near future prediction – such as one year in advance – is the main concern. One should nevertheless be aware that a more complete picture of the progressive pattern will only be visible as more real data are accumulated over time.

Lastly, the above models (1) and (3) are based on the assumption that the fitted residuals are normally distributed, which needs to be justified. While an apparent violation of normality can often be detected by simple graphical methods such as probability plot or QQ plot, numerical tests are necessary for subtle cases. Here four widely used procedures, namely, Kolmogorov-Smirnov (KS) test, Lilliefors (Lillie) test, Shapiro-Wilk (SW) test, and Anderson-Darling (AD) test were run and the corresponding results are presented in Table 6. For relevance and comparative powers of these tests for normality testing, one can refer to Razali and Wah (2011). For model (3), the null hypothesis (H_0) that the residuals are normally distributed is solidly affirmed as it passed all normality tests with sufficiently high p-values. On the other hand, the normality assumption is only marginally satisfied for model (1). When the significance level of the test, defined as the probability of the Type I error, is set at 0.01, model (1) can pass all the normality tests; but when the significance level is set at 0.05, it only passes KS test and fails all the rest (see Table 6 for detailed statistics).

This is to some extent understandable, as the CGWR process undergoes large fluctuations in the initial stage. Model (1) was created to capture only the main trend of the process in the first place, where the local fluctuations were not accounted until the trigonometric terms were introduced as in model (3). Because the variability of error induced by model (1) systematically decreases from large to small, the slight non-normality of it can be easily rectified. One convenient method for this purpose is to appeal to what is called the linearization transformation of random variables (Schabenberger and Pierce 2002; Chatterjee and Hadi 2012). In our case, the desired transformation is

$$Y = \ln(b_1 - E),$$

where E is entropy, the response variable of model (1), and $b_1 = 6.0222$ is the parameter appearing in model (1) defining the asymptotic upper bound of the entropic process of CGRW. A simple algebraic operation based on the above transformation yields a linear model of the following form:

$$Y = c_1 + c_2 t. \tag{4}$$

The rest of the original parameters of model (1) can be recovered from the parameters of the linearized model by $b_2 = e^{c_1}$ and $b_3 = c_2$. A Simple least square regression gives the best estimation of the transformed parameters as $c_1 = -1.2620$ and $c_2 = -0.0633$. Indeed, the transformed model defined by the above linear equation neatly satisfies the normality requirement, where the statistics of the corresponding normality tests are also tabulated in Table 6.

Table 6: Statistical tests for the normality of the models

		KS	Lillie	SW	AD
Model (1)	H_0 (0.01 significance)	accepted	accepted	accepted	accepted
	H_0 (0.05 significance)	accepted	rejected	rejected	rejected
	p-value	0.3474	0.0363	0.0137	0.0313
	Statistic	0.1387	0.1387	0.9321	0.8182
Model (1) Log-Transformed	H_0 (0.01 significance)	accepted	accepted	accepted	accepted
	H_0 (0.05 significance)	accepted	accepted	accepted	accepted
	p-value	0.9687	0.8230	0.3572	0.5978
	Statistic	0.0717	0.0717	0.9716	0.2987
Model (3)	H_0 (0.01 significance)	accepted	accepted	accepted	accepted
	H_0 (0.05 significance)	accepted	accepted	accepted	accepted
	p-value	0.9175	0.6750	0.4016	0.4216
	Statistic	0.0812	0.0812	0.9731	0.3672

One comment I would like to add is that the linearization procedure via logarithm transform only leads to the significance of normality for the modeling; it does not improve the model accuracy. In particular, model (1) is still relatively inferior compared to model (3), judged by the out-of-sample predicting errors, regardless it is expressed in terms of the original upper-bounded exponential function or the logarithm transformed linear function.

6 CONCLUDING REMARKS

Taking the CGWR as the test corpus, the current work investigates how lexical richness of a series of homogeneous texts evolves over a large time horizon. The dynamic complexity theory is shown to be a pertinent and valid foundation in the current study in the context of the existence conditions of homogeneous texts. It provides a relevant method for looking at the CGWR corpus as an open, dynamic, heterogeneous, and self-adaptive system. Additionally, the strong, distinctive, and mathematically describable properties exhibited in the associated time series of the entropic data are naturally hinted by the key theoretical implications of a dynamic complexity system. Although the base functions used in our modeling – a class of concave down exponential functions and a class of periodic trigonometric functions – seem rather simple, it takes a novel combination of them together with their interactions to produce an effective quantitative model. The models and results of the current work demonstrate that the dynamic complexity approach is not only metaphorically plausible, but is also conducive to rigorous quantitative conclusions.

A major limitation of the current study is the small size of the available data, resulting from the relatively short history of the CGWR practice. Given that the CGWR is an institutional writing process which is subject to the influence of the sociocultural environment in which it is embedded, it may be hasty to assume that the CGWR will reach its peak of lexical richness within fifty years since its inception. Because of this limitation, the models contained in the current study could have only provided a partial picture of an even larger evolving pattern which may not stand out until sufficient time has passed. For instance, it is possible that the periodicity of local fluctuations, described by the parameter b_5 in model (3), will not keep constant for an arbitrarily long time. In addition, it could be the case that the saturation state observed in the current paper is not the ultimate one

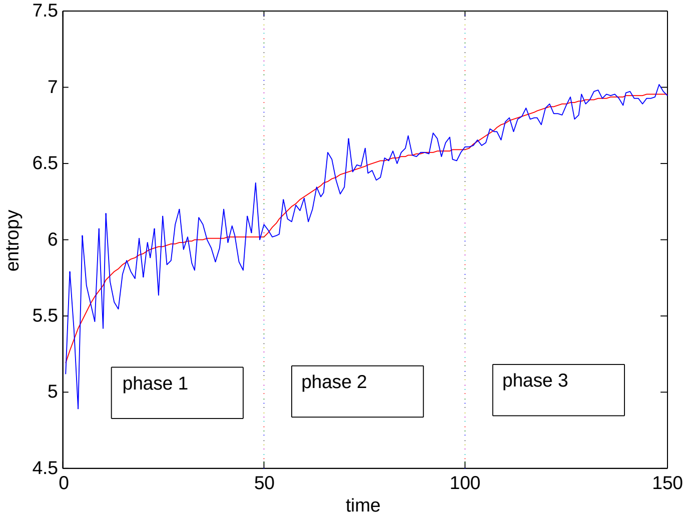


Figure 7: Illustration of multi-phase entropic growth

when the future evolution of the CGWR process is taken into consideration. Among other reasons, phase change is known to be a common characteristic of linguistic dynamics (Larsen-Freeman and Cameron 2008), implying the probableness that the saturation currently observed is only one of the multiple local saturations to come when the data is large enough. Figure 7 presents a simulated illustrative example where an entropic process undergoes three growth phases; each can be roughly estimated by model (3). This is in spite of the observation that all such three phases are again subordinated to a large-scale exponential decay model when interpolated together. More examples of multi-phase linguistic dynamics can be found in Larsen-Freeman and Cameron (2008), Verspoor *et al.* (2011), and Stachowski (2013), for instance.

The models provided in the current paper, when extrapolated backwards, can also provide useful hints to the pattern of language changes in historical linguistic studies. In particular, model (3), where it is appropriate to apply, tends to hint that language changed more dramatically in the farther past than in more recent times. To cite one example, the occurrence of paratactic constructions in written English such as left-dislocated NPs had undergone a roughly exponential decay during the years 950–1910 from Old English to Early Modern English and to Modern English, where the changes in the first 500 years were more vehement before being stabilized since about 1450 (van Kemenade and Los 2014). As an example in Chinese, the relative frequency of *ye3*, a sentence-final interjective marker which is often an emblematic of a Chinese text being classic, had seen gradual decrease from pre-10th century to 20th century, where the changes were more volatile and dramatic before 17th century (Shi 1989). This said, much depends here on the overall trending of the underlying process, there are cases where an exponential model is not suitable. The occurrence of unique Turkic glosses in Polish texts from 1388 to 1791 reported by Stachowski (2013) and the increase of the use of the English auxiliary *do* as a negative declarative demonstrated by Ellegård (1953) are such examples. Logistic functions, instead of exponential functions, should be used to best describe the respective linguistic phenomena, where a slow initial growth period is present before a more dramatic growth period emerges. In addition to model selection, the comparability and representativeness of the historical texts are also critically important when backward extrapolation is applied to infer language changes in the past. Caveats and pitfalls may arise because language data, when drawn from different historical periods, can be very inhomogeneous in dialect, genre, register, and sociolinguistic environment. For further technical precautions in using limited historical texts to extrapolate general pattern in the past one can refer to van Kemenade and Los (2014).

For future work, it is important to enrich the current research with similar empirical tests using other types of homogeneous texts in Chinese, so as to generalize the conclusions made in this study. Systematic differences in terms of the concavity of curve or parameter values or sharpness of the initial increasing phase of the curve might be detected when homogeneity changes across different corpora. Admittedly, however, the more challenging task will be how to account for such cross-corpora differences from pertinent theories, some of which can be more innately rooted in the mechanisms of language development. Dynamic complexity theory, generally concerned with the structural distribution from macro and inferential perspective, does serve as a substitute for causal effect analysis in specific linguistic fields.

In addition, testing of the models against homogeneous texts in languages other than Chinese might generate insightful comparisons. Given that Chinese and English are very different in many aspects, including orthographic form, syntactic rules, and semantic structure (Ku and Anderson 2003; Perfetti and Tan 1998), whether or not the lexical richness measures that were developed historically for alphabetical languages are readily applicable to Chinese as an orthographic language is a reasonable concern. As shown by Figure 1 of the current paper, the frequency distribution of the CGWR text (in log-log scale) can be very different than one might expect for English. Specifically, the frequency distribution of Chinese tends to exhibit a larger concavity after a certain rank of unique characters (typically in thousands) is reached, whereas that of English tends to progress with a more stable slope. This rank-frequency distributional difference between the two languages has been verified by recent empirical studies such as Chen *et al.* (2012). How this difference will affect the entropic process of English homogeneous texts and whether the pattern uncovered in the current study will equally hold for the counterpart in English will be a worthwhile future direction.

Another aspect desiring more fine-tuned investigation is the mechanism leading to the periodic, although modulated, fluctuations manifested in the entropic process of the CGWR texts. Possible approaches may include a careful examination of the recurrent sociolinguistic themes to which the CGWR sporadically refers. For example, strategic planning is a central characteristic of China's economy, where a top-down Five-Year Plan is developed by the government every five years to mobilize resources for identified priorities. It is then a legitimate question to ask whether a sort of correlation exists between such a recurring socioeconomic initiative and the observed periodic pattern in the entropic process of the CGWR text.

ACKNOWLEDGEMENT

The author is very grateful to three anonymous referees as well as the editors of JLM, whose comments and suggestions have been of great help in improving the earlier versions of this paper.

REFERENCES

Martin BAILYN (1994), *A Survey of Thermodynamics*, American Institute of Physics, New York.

Doug BEEFERMAN, Adam BERGER, and John LAFFERTY (1997), A model of lexical attraction and repulsion, in *Proceedings of the ACL*, pp. 373–380, Madrid, Spain.

Soren BISGAARD and Murat KULAHCI (2004), *Time Series Analysis and Forecasting by Example*, John Wiley & Sons, Hoboken, New Jersey.

Juliette BLEVINS (2004), *Evolutionary Phonology: The Emergence of Sound Patterns*, Cambridge University Press, Cambridge, MA.

Peter F. BROWN, Steven A. Della PIETRA, Vincent J. Della PIETRA, Jennifer C. LAI, and Robert L. MERCER (1992), An estimate of an upper bound for the entropy of English, *Computational Linguistics*, 18(1):31–40.

Samprit CHATTERJEE and Ali S. HADI (2012), *Regression Analysis by Example*, John Wiley & Sons, New York.

Qinghua CHEN, Jinzhong GUO, and Yufan LIU (2012), A statistical study on Chinese word and character usage in literatures from the Tang Dynasty to the present, *Journal of Quantitative Linguistics*, 19:232–248.

William CROFT (2008), Evolutionary linguistics, *Annual Review of Anthropology*, 37:219–234.

Scott A. CROSSLEY and Danielle S. MCNAMARA (2011), Shared features of L2 writing: Intergroup homogeneity and text classification, *Journal of Second Language Writing*, 20(4):271–285.

Etienne DENOUAL (2005), The influence of example-data homogeneity on EBMT quality, in *Proceedings of the Second Workshop on Example-Based Machine Translation*, pp. 35–42, Phuket, Thailand.

Alvar ELLEGÅRD (1953), *The Auxiliary Do: the Establishment and Regulation of Its Use in English*, Almqvist and Wiksell, Stockholm.

Robert FILDES (1992), The evaluation of extrapolative forecasting methods, *International Journal of Forecasting*, 8:81–98.

Dmitriy GENZEL and Eugene CHARNIAK (2002), Entropy rate constancy in text, in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 199–206, Philadelphia.

Stefen Th. GRIES (2006), Exploring variability within and between corpora: some methodological considerations, *Corpora*, 1(2):109–151.

Trevor HASTIE, Robert TIBSHIRANI, and Jerome FRIEDMAN (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York.

Douglas M. HAWKINS (2004), The problem of overfitting, *Journal of Chemical Information and Computer Sciences*, 44:1–12.

Scott JARVIS (2013), Capturing the diversity in lexical diversity, *Language Learning*, 63:87–106.

Victoria JOHANSSON (2008), Lexical diversity and lexical density in speech and writing: a developmental perspective, *Lund Working Papers in Linguistics*, 53:61–79.

Adam KILGARRIFF (2001), Comparing corpora, *International Journal of Corpus Linguistics*, 6(1):1–37.

Adam KILGARRIFF and Gregory GREFENSTETTE (2003), Introduction to the special issue on the web as corpus, *Computational Linguistics*, 29(3):333–348.

Andras KORNAI, Peter HALACSY, Viktor NAGY, Csaba ORZVECZ, Viktor TRON, and Daniel VARGA (2006), Web-based frequency dictionaries for medium density languages, in *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1–8, Trento, Italy.

Yu-Min KU and Richard C. ANDERSON (2003), Development of morphological awareness in Chinese and English, *Reading and Writing: An Interdisciplinary Journal*, 16(1):399–422.

Diane LARSEN-FREEMAN and Lynne CAMERON (2008), *Complex Systems and Applied Linguistics*, Oxford University Press, Oxford.

Namhee LEE and John H. SCHUMANN (2003), The evolution of language and the symbolosphere as complex adaptive system, paper presented at *the American Association of Applied Linguistics Conference*, Arlington, VA.

Brain MACWHINNEY (2007), A unified model, in P. ROBINSON and N. ELLIS, editors, *Handbook of Cognitive Linguistics and Second Language Acquisition*, Lawrence Erlbaum Associates, Mahwah, NJ.

Elinor MCKONE (1995), Short-term implicit memory for words and non-words, *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 21:1108–1126.

Paul MEARA (2006), Emergent properties of multilingual lexicons, *Applied Linguistics*, 27(4):620–644.

Charles A. PERFETTI and Lihai TAN (1998), The time-course of graphic, phonological, and semantic activation in Chinese character identification, *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 24:1–18.

Nornadiah M. RAZALI and Yap B. WAH (2011), Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests, *Journal of Statistical Modeling and Analytics*, 2(1):21–33.

Magnus SAHLGREN and Jussi KARLGREN (2005), Counting lumps in word space: density as a measure of corpus homogeneity, in *Proceedings of 12th Symposium on String Processing and Information Retrieval*, pp. 124–132, Buenos Aires, Argentina.

Oliver SCHABENBERGER and Francis J. PIERCE (2002), *Contemporary Statistical Models for the Plant and Soil Sciences*, CRC Press, New York.

Claude E. SHANNON (1951), Prediction and entropy of printed English, *Bell System Technical Journal*, 30:50–64.

Ziqiang SHI (1989), The grammaticalization of the particle le in Mandarin Chinese, *Language Variation and Change*, 1:99–114.

Joseph. A. SMITH and Colleen KELLY (2002), Stylistic constancy and change across literary corpora: Using measures of lexical richness to date works, *Computers and the Humanities*, 36:411–430.

Michael SPIVEY (2007), *The Continuity of Mind*, Oxford University Press, Oxford.

Kamil STACHOWSKI (2013), The influx rate of Turkic glosses in Hungarian and Polish post-mediaeval texts, in R. KÖHLER and G. ALTMANN, editors, *Issues in Quantitative Linguistics*, pp. 100–116, RAM-Verlag, Lüdenscheid.

Sune V. STEFFENSEN and Alwin FILL (2014), Ecolinguistics: the state of the art and future horizons, *Language Sciences*, 41(6):6–25.

Benedikt SZMRECSANYI (2005), Language users as creatures of habit: A corpus-based analysis of persistence in spoken English, *Corpus Linguistics and Linguistic Theory*, 11:113–150.

Ans VAN KEMENADE and Bettelou LOS (2014), Using historical texts, in D. SHARMA and R. PODESVA, editors, *Research Methods in Linguistics*, pp. 216–231, Cambridge University Press, Cambridge.

Marjolijn H. VERSPOOR and Heike BEHRENS (2011), Dynamic systems theory and a usage-based approach to second language development, in M. VERSPOOR, K. DE BOT, and W. LOWIE, editors, *A Dynamic Approach to Second Language Development: Methods and Techniques*, pp. 25–38, John Benjamins, Amsterdam.

Marjolijn H. VERSPOOR, Kees DE BOT, and Wander LOWIE, editors (2011), *A Dynamic Approach to Second Language Development: Methods and Techniques*, John Benjamins, Amsterdam.

William S-Y. WANG (1979), Language change: a lexical perspective, *Annual Review of Anthropology*, 8:353–371.

Jeffrey S. WICKEN (1987), Entropy and information: suggestions for common language, *Philosophy of Science*, 54:176–193.

Dick R. WITTINK (1988), *The Application of Regression Analysis*, Allyn and Bacon, Boston, MA.

EXTERNAL REVIEWERS 2014–2015

The mainstay of any peer-reviewed journal are its reviewers, and JLM is no exception here. Each paper is reviewed by at least 3 carefully selected reviewers, including at least one representing the JLM Editorial Board. To increase reviewing anonimity, we do not give the names of the 28 JLM EB reviewers, but we would like to heartily thank them for their hard and timely work. We also express our sincere gratitude to the following 79 external reviewers for papers reviewed during 2014–2015:

Elena Anagnostopoulou

Avery Andrews

Doug Arnold

Denis Béchet

Emily Bender

Paul Boersma

Igor Boguslavsky

Johan Bos

Miriam Butt

Marie Candito

John Carroll

Robin Cooper

Ann Copestake

Benoît Crabbé

Berthold Crysmann

Peter Culicover

Michael Cysouw

Eric De La Clergerie

Huy-Linh Dao

Łukasz Dębowski

Magdalena Derwojedowa

Danh Thành Do-Hurinville

Denys Duchier

Maciej Eder

Nathaniel Filardo

Martin Forst

Anette Frank

Dafydd Gibbon

Joanna Golińska-Pilarek

Mike Hammond

Jeffrey Heinz

Mans Hulden

Adam Jardine

Sylvain Kahane

Ron Kaplan

Roni Katzir

Andre Kempe

Elma Kerz

Adam Kilgarriff

Gregory Kobele

Jacek Koronacki

Seth Kulick

Joseph Le Roux

William Leben

Adam Lopez

Wolfgang Maier

Andreas Maletti

Krzysztof Marasek

Montserrat Marimon

Nazarre Merchant

Marcin Miłkowski

Lawrence Moss

Andrew Nevins

Maciej Ogrodniczuk

Yannick Parmentier

Nina Pawlak

Adam Pawłowski

Francis Pelletier

Sylvain Pogodalla

Alan Prince

Siamak Rezaei

Jason Riggle

Strahil Ristov

Graeme Ritchie

Josef Ruppenhofer

Benoît Sagot

Djamé Seddah

Kaius Sinnemäki

Richard Sproat

Harold Torrence

Lynette Van Zijl

Cristina Vertan

Jürgen Wedekind

Michael Wiegand

Stephen Winters

Anssi Yli-Jyrä

Sina Zarriess

Amir Zeldes

Bartosz Ziółko