A Workshop on

Ideas and Strategies for Multilingual Grammar Development

Emily Bender, Dan Flickinger, Frederik Fouvry and Melanie Siegel (editors)

25–29 August 2003

Taking place during the
Fifteenth European Summer School for Logic,
Language and Information
18–29 August, Vienna, Austria

Preface

This volume contains the papers presented at the workshop on Multilingual Grammar Engineering, held from the 25th to 29th August 2003 in conjunction with the European Summer School for Logic, Language and Information 2003, 18-29 August in Vienna (ESSLLI 2003).

Many broad-coverage grammars have been developed over the past decade that represent deep linguistic analysis in several frameworks, including Head-driven Phrase Structure Grammar (HPSG), Lexical-Functional Grammar (LFG), and Lexicalized Tree Adjoining Grammar (LTAG). With a few notable exceptions, the largest and richest such grammars have been developed independently, for one language at a time, even when grammars for other languages are being developed within the same general framework. Recently, however, more efforts have begun to parallelize the grammar development process across languages. In this workshop, participants address the issue of building a methodology for parallel grammar development in linguistically rich frameworks. This methodology should guide the definitions of common formats, procedures, development tools, grammar components, and documentation practice, as well as standardized evaluation methods.

Through papers addressing formal devices and the use of multilingual grammars, multilingual grammar writing techniques, acquisition and extraction of multilingual grammars and grammars of particular languages built within the framework of multilingual grammar engineering efforts, workshop participants address the following topics:

- methodology for multilingual, broad-coverage, grammar development within linguistically rich frameworks such as those using unification-based grammars;

- guidelines for grammar writers either in the initial stages of development or as long-term best practices;

- organization of a grammar in layered, reusable structures;

- strategies for adapting existing resources such as taggers or morphological analyzers.

The workshop aims to draw attention to the multilingual aspect of grammar writing and to the necessity for setting up methods of standardization and support for new grammars in different languages. We hope these goals can be achieved.

Emily M. Bender
Dan Flickinger
Frederik Fouvry
Melanie Siegel

Programme Committee

Anne Abeillé, Université Paris 7, France
Emily M. Bender, Stanford University, USA
Francis Bond, NTT Japan
Dan Flickinger, CSLI Stanford, USA
Frederik Fouvry, Saarland University, Germany
Tracy Holloway King, Palo Alto Research Center, USA
Stephan Oepen, CSLI Stanford, USA
Melanie Siegel, Saarland University, Germany

Table of Contents

A grammar component for semantic classes of nominals in HPSGs

Kaja Borthen
Department of language and communication studies
Norwegian University of Science and Technology
kaja.borthen@hf.ntnu.no

Abstract

In this paper I propose a grammar component for defining and picking out semantic classes of nominals across languages within the HPSG framework, thus providing the basis for accounting for grammatical phenomena that bear on this issue. I introduce a semantic object that includes the referential index, or discourse referent, of a nominal, as well as different kinds of semantic features associated with this index (i.e. cognitive status, referentiality, and partitivity). This semantic object can be seen as the discourse referent as it is "dressed up" in different kinds of costumes that make it recoverable and recognizable in sentences and in discourse. Semantic classes of nominals correspond to subtypes of this "dressed up" discourse referent.

1 Some phenomena

Many grammatical phenomena are sensitive to semantic classes of nominals, of which I will here mention five that appear in Norwegian.

1) Norwegian unaccented, definite pronouns (so-called "light" pronouns), differ from all other nominals in Norwegian (including unaccented *indefinite* pronouns) in that they occupy the position in between a main verb and a sentential adverb. This is illustrated in (1).

(1) Jo så ham/*hám/*gutten/*en/*en gutt ikke.
Jo saw him/*HIM/*boy-DEF/*one/*a boy not
'Jo didn't see him.'

One apparent characteristics that distinguishes "light pronouns" from all other nominal constituents in Norwegian is that they signal that their associated discourse referent is in current centre of attention (see e.g. Fretheim et al., 1997). It is therefore reasonable to treat this word order phenomenon as one sensitive to semantic classes of nominals, as in Haugereid (2001).[1]

2) Just like in English, the postverbal nominal in Norwegian existential sentences has to be *weak* in the sense of Milsark (1977) (see Vangsnes, 1994). This means that they have to be indefinite, and that they cannot be partitive, referential, or universal. This is illustrated in (2).

(2) a. Det er en hund/en i hagen.
there is a dog/one in garden-DEF
'There is a dog/one in the garden.'
b. Det er *hunden/*en av hundene/*den/*alle hunder i hagen.
there be-PRES *dog-DEF/*one of dogs-DEF/ *it/*all dogs in garden-DEF

The class of nominals that can appear posverbally in existential sentences is most naturally defined semantically rather than syntactically.

3) Left-dislocation is possible with all types of definite nominals in Norwegian, except unaccented definite pronouns, as illustrated in (3a)

[1] No other syntactic categories than nominals can appear in the "light pronoun"-position illustrated in (1). Thus, for this phenomenon, there is no need to introduce a feature or synsem type that can apply across syntactic categories, as proposed in Abeillé and Godard (to appear) for French.

below. As illustrated by the contrast between (3b) and (3c), indefinites cannot be left-dislocated in this type of left-dislocation, unless the phrase has sufficient descriptive content to support that the speaker has a particular individual in mind.[2]

(3) a. Bilen/den bilen/dén/*den, dén liker jeg.
car-DEF/that car-DEF/that/*it, that like I
'The car/that car/that, I like it.'
b. *En bil/*en av bilene/*en, dén liker jeg.
*a car/*one of cars-DEF/*one, I like it
c. En gutt i klassen min, han sovnet.
a boy I in class-DEF mine, he fell-asleep
'A boy in my class, he fell asleep.'

To sum up (a bit loosely at this point), this construction calls for a way of defining a class of nominals whose associated discourse referent has a "not-too-high" and "not-too-low" degree of referential givenness.

4) As argued in Borthen (2003), Norwegian bare singulars have a different semantics than corresponding nominals with the indefinite article. Among other things, they cannot be referential. One consequence of this is that these phrases can never combine with so-called specificity adjectives such as *spesifikk* ('specific') and *viss* ('certain'), as shown in (4).

(4) Per ønsker seg ny/stor/*viss/*spesifikk bil.
Per wants REFL new/big/*certain/*specific car
'Per wants a new/big/certain/specific car.'

5) As illustrated by the contrast between (5a) and (5b), Norwegian adjectives achieve different forms depending on whether they occur in semantically definite or indefinite nominals. The form of an adjective depends on what kind of determiner it combines with, but when adjectives appear in determinerless phrases, as in (5c) and (5d), it is the form of the adjective that determines how the phrase is interpreted semantically. This is illustrated in (5cd), as the object in (5d) is semantically indefinite, whereas the subject in (5e) is semantically definite.

(5) a. Per har en pen/*pene hund.
Per has a pretty/*pretty-DEF dog
'Per has a pretty dog.'
b. Min pene/*pen hund bjeffet.
the pretty-DEF/*pretty dog barked
'My pretty dog barked.'
c. Per har pen hund.
Per has pretty dog
'Per has a pretty dog.'
d. Pene hund, kom hit!
pretty-DEF dog, come here
'Pretty dog, come here!'

This suggests that adjectives with "indefinite" form are specified as being part of semantically indefinite nominals, whereas adjectives with "definite" form are specified as being part of semantically definite nominals.

More generally, there exist various phenomena in a number of languages that have been argued to be sensitive to semantic classes of nominals. Existential sentences, antecedent-contained deletion, and extraction in English, case marking in Turkish, and extraction and scrambling in German are some such phenomena discussed in Diesing (1991), for instance.

2 Lack of consensus

The HPSG literature and the different broad-coverage grammars that have been developed lately have so far not landed on a convention for representing semantic classes of nominals. For instance, definiteness (e.g. DEF *definiteness*) is commonly represented as a syntactic feature in nominals (see e.g. van Eynde, 2002), adjective inflection of the type illustrated in (4) is handled by a syntactic feature (i.e. DECL *bool*) in e.g. Neville (2000), ACTIVATED *bool*, is a feature on *ctxt* in the current English Resource Grammar (Bender, Flickinger, and Oepen (version 0.4), 2003)), SPECIFIC *bool* is assumed to be a feature on *content* in Asudeh and Mikkelsen (2000), and the current Norwegian HPSG grammar NorSource (Hellan and Haugereid, 2003), has a head feature on nominals reflecting the distinction between "light" and "heavy" pronouns; i.e. HEAVY *bool*.

Both from a theoretical point of view, and from the point of view of parallel grammar development, it must be regarded a desirable tacit to minimize the number of features introduced to represent semantic classes of nominals, and to standardize their position in the grammar archi-

[2] The restriction against nonreferential left-dislocated indefinites only holds in the case where the pronoun that initiates the nucleus clause is a *token pronoun*, as in (3), as opposed to a *type anaphor*. We will return to this issue in section 4 below.

tecture. In this paper, I will therefore propose a general approach to defining semantic classes of nominals within HPSG-style grammars.

3 Three semantic features

I assume we need three semantic features to account for the Norwegian data in section 1, and it is my strong conviction that these features will be useful for accounting for a wide range of similar phenomena in other languages as well. The three features correspond to the notions *cognitive status*, *referentiality*, and *partitivity*, as described below.

3.1 Cognitive status

Based on empirical studies in five different languages, Gundel et al. (1993) conclude that determiners and pronouns constrain possible interpretations of nominal forms by conventionally signaling one of six cognitive statuses that the intended referent is assumed to have in the mind of the addressee. The cognitive statuses are related in the Givenness Hierarchy in (6) below, and are to be understood as necessary and sufficient conditions for the appropriate use of the forms that they are mapped up with.

(6) <u>The Givenness Hierarchy</u>:

in focus	{*it*}
↓	
activated	{*that, this, this N*}
↓	
familiar	{*that N*}
↓	
uniquely identifiable	{*the N*}
↓	
referential	{*indefinite this N*}
↓	
type identifiable	{*a N*}

The different statuses that Gundel et al. propose are illustrated by the text fragments in (7).

(7) i. I couldn't sleep last night.
   ii. a. A dog (next door) kept me awake.
       b. This dog (next door) kept me awake.
       c. The dog (next door) kept me awake.
       d. That dog (next door) kept me awake.
       e. That kept me awake.
       f. It kept me awake.

The form *a dog* in (7a) is appropriate only if the associated discourse referent has the status *type identifiable*, which means that the hearer is able to identify the type of thing referred to. As for *this dog* in (7b), this form encodes the status *referential*. This means not only that the discourse referent is type identifiable, but also that the speaker has a particular individual in mind, which in turn means that the hearer is supposed to associate a representation of the discourse referent by the time the sentence is processed. The expression *the dog (next door),* in (7c) signals that the associated discourse referent is at least *uniquely identifiable*. This means either that the hearer is familiar with the dog already, or that the descriptive content in the phrase is so rich that the hearer can create a uniqe representation of the discourse referent based on this description alone. The form *that dog (next door)* in (7d) signals the more restrictive cognitive status *familiar*, which means that the hearer already has a representation of the associated discourse referent in memory. *That*, in (7e) encodes the cognitive status *activated*, which means that the discourse referent has recently been mentioned or retrieved from extralinguistic context, which entails that the hearer has a representation of the associated discourse referent in short-term memory. And finally, the personal pronoun *it* in (7f) signals the most restrictive cognitive status, namely *in focus*, which means that the discourse referent is currently in the center of attention.

The cognitive statuses in (6) are intended to be relevant for accounting for appropriateness and interpretation of nominal forms in discourse, which is indeed a context-dependent matter. One of the main claims that I want to make in this article is that the very same categories are also relevant for grammatical phenomena like those in (1)-(5); we just need to modify the hierarchy of cognitive statuses a bit.

As indicated by the arrows in (6), the higher cognitive statuses entail the lower ones. So, if something is in focus, it is also activated, familiar, uniquely identifiable, referential, and type identifiable. This means that any form that signals a certain minimal cognitive status for its discourse referent can also be used if a higher cognitive status is actually obtained. For instance, it is predicted that the definite article can be used even if the associated discourse referent is in focus of attention, since being in focus entails being uniquely identifiable.[3] Although plausible when used to predict appropriateness of nominal forms in discourse, the entailment relations between the statuses in (6) make it hard to use the Givenness Hierarchy to make predictions about context-independent grammatical phenomena.

If we translate the Givenness Hierarchy in (6) into a type hierarchy of the conventional HPSG format, we get the following:

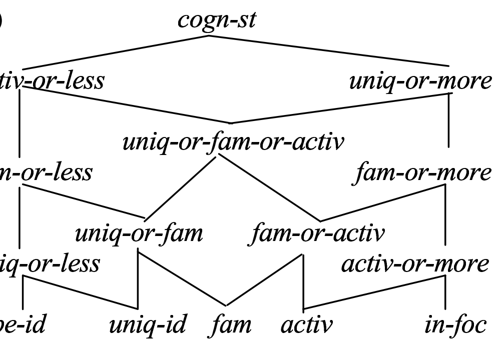
(8) *type-id*
|
(referential)[4]
|
uniq-id
|
fam
|
activ
|
in-foc

This type hierarchy correctly expresses the entailment relations in (6). Like in (6), the cognitive status type *in-foc* is the most restrictive one, since no other types inherit from it, and it follows from (8) that if something has the cognitive status specification *in-foc*, it is also *activ*, *fam*, *uniq-id*, *referential*, and *type-id*, as daughters inherit all the properties of their mother(s).

The problem with the type hierarchy in (8) is that there is only one line of dominance, which means (since mothers and daughters are always compatible with each other) that all the types in (8) are compatible. Imagine that we attempt to use the type hierarchy in (8) to ensure that the "light pronoun word order" in (1) is only possible with unaccented definite pronouns. A plausible approach seems to be to restrict unaccented, definite pronouns to have the cognitive status specification *in-foc*, and then restrict the "light pronoun word order" to license only nominals marked as *in-foc*. But this in fact makes no predictions with respect to nominal forms. Since the status *in-foc* is compatible with all other statuses in (8), all types of nominals are in fact licensed. The effect of having an indefinite nominal in the "light pronoun" position will only be that this indefinite gets the cognitive status value *in-foc*.

Thus, for grammar-internal, purposes it is necessary to introduce absolute oppositions between (classes of) cognitive statuses, as in (9) below. With this hierarchy we are able to pick out individual (incompatible) cognitive statuses (i.e. *type-id* vs. *uniq-id* vs. *fam* vs. *activ* vs. *in-foc*), as well as groups of cognitive statuses that form natural semantic classes (e.g. *activ-or-less*, *uniq-or-fam-or-activ*, and *activ-or-more*). There are no "discontinuous" groups of cognitive statuses in (9), as I assume that such classes are not linguistically relevant.

(9)



The hierarchy in (9) is rather powerful. As we will see in section 5 below, this one feature COGN-ST *cogn-st* does the job of the four independently justified features HEAVY *bool*, ACTIVATED *bool*, DECL *bool*, and DEF *definiteness* mentioned in section 2 above. More generally, it prevents the introduction of unnecessarily many boolean features all having to do with cognitive status.

The feature COGN-ST *cogn-st* and the hierarchy in (9) can also take off some of the burden possibly carried by (hierarchies of) determiner relations. Determiner relations are anyway not particularly suitable for accounting for the external distribution of noun phrases, since they are elements on (difference) lists together with other relations introduced in a nominal phrase. Since determiner relations are (usually) not even assumed to be the "key" relation in a nominal phrase, they are not kept track of, which is a pre-

[3] The fact that some forms are typically *not* used with higher cognitive statuses, can be accounted for by Gricean implicatures, according to Gundel et al..

[4] I take Gundel et al.'s status *referential* out of the hierarchy of cognitive statuses and rather insert it as a separate feature (see sections 3.2 and 4).

requisite for exploiting them for noun-phrase-external purposes. Furthermore, whereas the present approach invites to a unified analysis of morphological, lexical, and constructional contributions to (certain aspects of) the meaning of nominals, an approach which represents cognitive statuses in terms of relations is not as suitable for accounting for such morphological effects. According to The HPSG Grammar Matrix, Bender (2000) (version 0.1), for instance, affixes are not assumed to introduce relations.

3.2 Referentiality

Differently from cognitive status, which is a hearer-oriented property of discourse referents, referentiality (i.e. REFERENTIAL *bool*) is here intended to be a speaker-oriented notion. That is, whereas cognitive status has to do with signaled attention state of a discourse referent in the *hearer*, referentiality has to do with whether or not the speaker (or some individual referred to by the speaker) has a particular individual in mind, to which the speaker intends to refer by the given nominal.

The choice of taking referentiality out of the Givenness Hierarchy is motivated by some nominal phrases in Norwegian that consist of an adjective with definite form and a noun with no definite suffix (see Borthen, 1998, for more details). Consider the following data:

(10)a. Beste elev vant.
best-DEF student won
'The best student won.'
b. Neste kunde skal få en premie.
next-DEF customer shall get an award
'The next customer will receive an award.'

The determinerless phrases in (10) are definite in the sense that their associated token discourse referent is supposed to be uniquely identifiable to the hearer, but they can never be referential in the sense that the speaker knows the referent and intends to refer to exactly this referent. The one who receives an award in (10b), for instance, is whoever turns out to be the next customer; it can't be a certain person the speaker has in mind. This indicates that it might be fruitful to cross-classify referentiality and cognitive statuses.

3.3 Partitivity

Partitivity is the last semantic feature that I want to introduce here. Obviously, a phrase like *one of the dogs* is partitive, and a phrase *N* is partitive if it means *one of the N*s. Partitivity is related to cognitive status, since it has to do with referential givenness. But like referentiality, it cannot be reduced to it, since both semantically definite and indefinite nominals can be partitive, as illustrated by the definite phrase *den største av jentene* ('the biggest of girls-DEF').

4 A refined inventory of HPSG indices

We have now introduced the features I assume are needed to account for the kind of phenomena illustrated in section 1. The question to be answered now is where in the feature geometry these features should be introduced.

Pollard and Sag (1994) propose that the semantic index of a nominal carries agreement information. As shown in (11), the features PER *per*, NUM *num*, and GEND *gend* (PNG *png*) are stated on the type *ref,* which corresponds to the referential index, or discourse referent, of a nominal.

$$(11)\quad \left[\text{SS.LOC.CONT.INDEX}\ \begin{bmatrix} \textit{ref} \\ \text{PER}\ \textit{per} \\ \text{NUM}\ \textit{num} \\ \text{GEND}\ \textit{gend} \end{bmatrix}\right]$$

This step provides an efficient way of accounting for (certain types of) agreement, since coindexation entails agreement.

Intuitively, also cognitive status, referentiality, and partitivity are features that are closely associated with a discourse referent. However, it would be counter intuitive and lead to wrong predictions to assume that cognitive status is a feature on *ref,* since one presumably does not want to assume that coreference entails sameness of cognitive status. In the case of discourse anaphora, for instance, it is not the case that the antecedent has to signal the same cognitive status as the anaphor does. For this and other reasons, I propose to introduce a linguistic object that includes the discourse referent as well as features representing cognitive status, referentiality, and partitivity, i.e. *dressed-ref.* The resulting representation is given in (12).

(12)

$$\left[\begin{array}{ll} \text{SS.LOC.CONT.INDEX} & \left[\begin{array}{l} \textit{dressed}-\textit{ref} \\ \text{REF-IND} \left[\begin{array}{l} \textit{ref} \\ \text{PNG } \textit{png} \end{array}\right] \\ \text{COGN-ST } \textit{cogn-st} \\ \text{REFERENTIAL } \textit{bool} \\ \text{PARTITIVE } \textit{bool} \end{array}\right] \end{array}\right]$$

REF-IND *ref* now represents the discourse referent, or index, of a sign,[5] whereas the INDEX value *dressed-ref* (metaphorically speaking) is the discourse referent as it is dressed up at a given reference act in different kinds of costumes at that make it recognizable and recoverable in sentences and in discourse.

The path to the "true" index of a nominal sign is now longer than in Pollard and Sag (1994), which - other things being equal - is a drawback. Let me therefore present some arguments in favor of the present feature architecture as opposed to other alternatives that might seem plausible.

First, an approach where some or all of the three newly introduced features are syntactic features, is not desirable, first of all because the semantic reflexes then have to be stated separately.

To locate the features under consideration on CONT *cont* is better than treating them as syntactic features, but does not lead to an analysis as parsimonious as the present one, since *cont* includes e.g. lists of relations, which are presumably not relevant for stating semantic classes of nominals. Furthermore, the features on *dressed-ref* constitute a natural class, as structure sharing of all these features appears relatively frequently, as will be illustrated in section 5. But the most decisive argument against introducing the three new features on *cont*, as I see it, has to do with the data in (13).

(13) Bil, det har jeg allerede.
car(MASC), that-NEUT have I already
'A car, I already have that.'

As mentioned in section 1, Norwegian left-dislocated indefinites have to be referential when the duplicating pronoun which initiates the nucleus clause is a token pronoun, i.e. a pronoun that refers to the same token as its antecedent, which is the case in (3). In those examples, the sentence topics are understood to be tokens, or individuals. In (13), on the other hand, the duplicating pronoun is the type anaphor *det,* which is indefinite in the sense that it introduces a new discourse referent of the same type as its antecedent. In this case the topic of the sentence is the type of thing *car*, and the left-dislocated indefinite does ***not*** need to be referential. In fact, it cannot be. How can we explain this?

In Borthen (2003) I argue that an indefinite nominal introduces two discourse referents, i.e. a type discourse referent and a token discourse referent. Whereas the token discourse referent (which corresponds to what we have so far called the discourse referent) is associated with the cognitive status specification *type identifiable*, a type discourse referent will always be at least familiar. After all, if a hearer knows the meaning of some noun, then the the type of thing it is used to refer to, i.e. its type discourse referent, is stored in memory as a familiar entity. Now, if we assume that it is the discourse referent that corresponds to the *topic* of a left-dislocation that has to be either uniquely identifiable or referential, then the topics in (3) (i.e. the token discourse referents) and the topic in (13) (i.e. the type discourse referent) have the required degree of referential givenness.

Even though space limits prohibit us from going into details about how the type discourse referent might be represented, the relevant data show that there are reasons to not have COGN-ST as an attribute on *cont*, since that would leave it open whether it is the token discourse referent or the type discourse referent that has the given cognitive status. Rather, I assume that COGN-ST *cogn-st* is a feature that is connected to any type of discourse referent.

Finally, a last alternative position for some or all of the three newly introduced features is on *ctxt.* In Pollard and Sag (1994), *ctxt* is supposed to host linguistic information that is context de-

---

[5] In Borthen (2003) I assume yet another feature on *ref*, namely DR *dr*, which is the *real* discourse referent of a nominal, stripped from all agreement features. With this feature, one can represent coreference also in the cases where anaphora that do not agree with their antecedents w.r.t. natural gender, person, or number (see Pollard and Sag (1994)). For space considerations, we will not consider that feature here.

pendent. Clearly, cognitive status as presented by Gundel et al. (1993) is a context-dependent notion, and many nominals are underspecified for partitivity and referentiality. However, the goal of this work is just to account for grammatical phenomena that bear on semantic classes of nominals that is signalled by a certain form, not phenomena that bear on *actual* cognitive status of a discourse referent. For instance, the light pronoun word order in (1) is only possible with nominals that *signal* the minimal cognitive status *in-foc* (i.e. unaccented definite pronouns); it is ***not*** possible with e.g. a nominal with the definite article whose discourse referent actually happens to be in focus of attention. The same holds for the other phenomena mentioned in section 1. Phenomena that do rely on actual cognitive status should indeed be handled by way of appropriate features on *ctxt*, though. For instance, Gundel et al.'s original hierarchy of cognitive statuses, (3), as intended to account for appriateness in discourse, belong to the context-dependent component of the grammar. Notice, though, that even though this would mean to introduce a new attribute, the type hierarchy of cognitive statuses in (9) can be used for that purpose as well.

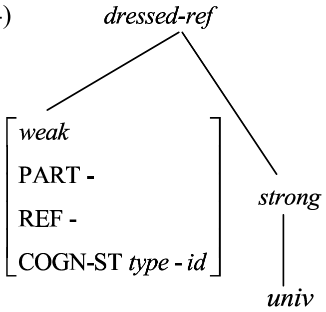
5 Analyses

Let's now return to the phenomena outlined in section 1 and provide the semantic categories needed to account for them.

1) The "light pronoun" word order in Norwegian will be restricted to license nominals that have the cognitive status specification *in-foc*, which is a cognitive status that only definite unaccented personal pronouns are compatible with. This approach is preferrable to the use of the feature HEAVY *bool* in the NorSource grammar, since cognitive status is a semantic aspect of nominals that we want to represent anyway. We can in other words get rid of an unnecessary feature.

2) For the distribution of nominals in Norwegian existential sentences, it might be useful to define the type *weak*, as in (14). If the semantic subject in existential sentences is specified as having the INDEX value *weak*, definites will be excluded since their COGN-ST value would be incompatible with the type *type-id*, partitive indefinites will be excluded, as they are presumably marked as PART +, and all indefinites in existential sentences will be marked as nonreferential. For reasons that I will not discuss here, I do not represent universal interpretations with a feature, but rather assume that universal determiners have the index value *univ*, which is incompatible with the type *weak.* For languages like Turkish, where one wants to refer to strong nominals directly (since nouns with accusative case are strong), and not only weak ones, the type *strong* in (14) needs to be enriched, of course.

(14)



3) As for left-dislocation, we have the features we need in order to state the relevant restrictions. The fact that definite pronouns, that are specified as *in-foc*, are excluded, can be captured by stating that the topic's cognitive status has to be *activ-or-less*, which is incompatible with *in-foc* according to (9). Thus, the feature COGN-ST *cogn-st*, overlaps with the feature ACTIVATED *bool* in the English Resource Grammar, which is used to prevent definite pronouns from being topicalized. As for left-dislocation of indefinites, stating the precise generalizations is complicated, since they depend on whether or not the topic of the sentence is the token discourse referent or the type discourse referent. But even though a full analysis of this is not presented here, we have at least ensured that each restriction can in principle be stated, since referentiality is represented through the feature REF *bool*, and since COGN-ST *cogn-st* has been argued to be a feature connected to discourse referents, and not constituents as such.

4) The fact that Norwegian bare singulars cannot combine with specificity adjectives, can easily be accounted for. Bare singulars are simply marked as REF -, whereas specificity adjectives will be marked as REF +. Assuming that adjectives and nouns share INDEX value (see (15)

below), this leads to the desired predictions. Differently from Asudeh and Mikkelsen (2000), who assume that the corresponding feature SPECIFIC *bool* belongs to *cont*, I need not state coreference and compatibility w.r.t. referentiality separately.

5) Finally, Norwegian adjectives with definite inflection will have the cognitive status specification *uniq-or-more*, as illustrated in (15), whereas indefinite ones will be *type-id*. Given that adjectives share INDEX value with the noun they modify, and assuming that nouns share INDEX value with the determiner they combine with, we predict that adjectives with indefinite form are part of indefinite nominals, whereas adjectives with definite form are part of definite ones.

(15)

$$\begin{bmatrix} \textit{local} \\ \text{CAT.HEAD} \begin{bmatrix} \textit{adj} \\ \text{MOD} \left\langle \begin{bmatrix} \text{LOC.CONT.INDEX } \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{CONT.INDEX } \boxed{1} \begin{bmatrix} \text{COGN-ST } \textit{uniq - or - more} \end{bmatrix} \end{bmatrix}$$

A syntactic approach to account for this type of adjective inflection, as proposed for equal patterns in Swedish and Danish (see Börjars, 1998, and Neville, 2000), is not as appealing, as it fails to account for the observed semantic effects when adjectives appear in determinerless phrases. Compared to the analysis of Neville (2000), we furthermore get rid of an unnecessary syntactic feature DECL *bool*.

To sum up, then, the introduction of the three features COGN-ST *cogn-st*, PARTITIVE *bool*, and REFERENTIAL *bool*, provide a promising basis for accounting for a range of grammatical phenomena sensitive to semantic classes of nominals. Especially the feature COGN-ST *cogn-st*, which draws heavily on insights made by Gundel et al. (1993), has been shown to be particularly powerful. The choice of letting these features being semantic features on the same linguistic type gives relatively efficient analyses of the phenomena under consideration, and the fact that the features are connected to discourse referents rather than constituents seems appealing both from an empirical and intuitive perspective, although more research on this topic is called for.

References

Abeillé, Anne & Danièle Godard. French word order and Lexical Weight. To appear in R. Borsley (ed.) *Syntactic categories*. Academic Press.

Asudeh, Ash. & Line H. Mikkelsen. 2000. Incorporation in Danish: Implications for interfaces. Cann, R., C. Grover, and P. Miller: *A Collection of Papers on Head-driven Phrase Structure Grammar*. Stanford University.

Borthen, Kaja. 1998. Linguistic evidence prompting the linguistic necessity of Donnellan's referential-attributive distinction? The use of bare superlatives in Norwegian. Manuscript.

Borthen, Kaja. 2003. Norwegian bare singulars. Ph.D. thesis. Norwegian University of Science and Technology.

Börjars, K. 1994. *Feature Distribution in Swedish Noun Phrases*. Ph.d. dissertation, Department of Linguistics, University of Manchester.

Diesing, Molly 1991. *Indefinites.* MIT Press.

Fretheim, Thorstein, Wim van Dommelen, and K. Borthen. 1997. Linguistic Constraints on Relevance in Reference Resolution. *CLS 33. Papers from The Main Session 1997.* Eds.: K. Singer, R. Eggert, G. Anderson. pp. 99-113.

Gundel, Jeanette., Nancy Hedberg, Ron Zacharski. 1993. Cognitive status and the form of referring expressions in discourse. *Language* 69: 274-307.

Haugereid, Petter. 2001. "Light pronouns in Norwegian: a syntatic-semantic declarative approach." Hovedoppgave. Department of Linguistics, NTNU, Trondheim.

Milsark, G.L. 1977. Toward an Explanation of Certain Peculiarities of the Existential Construction in English. In *Linguistic Analysis* 3, 1-29.

Neville, Ann 2000. *An HPSG Account of Danish Prenominals.* Ph.D. dissertation, Department of General and Applied Linguistics, Copenhagen University.

Pollard, Carl and Ivan Sag. 1994. *Head-Driven Phrase Structure Grammar.* The University of Chicago Press, Chicago & London.

Van Eynde, Frank. 2002. Prenominals and pronominals in Dutch. Manuscript, September 2002.

Vangsnes, Øystein. 1994. The syntactic source and semantic reflexes of the definiteness effect. Cand philol. thesis, University of Bergen, Dept. of Linguistics and Phonetics.

The Feature Space in Parallel Grammar Writing

Miriam Butt
Centre for Comp. Linguistics
UMIST
PO Box 88
Manchester M60 1QD UK
mutt@ccl.umist.ac.uk

Martin Forst
IMS
Universität Stuttgart
Azenbergstr. 12
70174 Stuttgart GERMANY
forst@ims.uni-stuttgart.de

Tracy Holloway King
NLTT/ISTL
Palo Alto Research Center
3333 Coyote Hill Rd
Palo Alto, CA 94304 USA
thking@parc.com

Jonas Kuhn
Dept. of Linguistics
The University of Texas at Austin
1 University Station, B5100
Austin, TX 78712-1196 USA
jonask@mail.utexas.edu

Abstract

This paper discusses the methodology and tools applied in the Parallel Grammar project (ParGram) to support consistency and parallelism of linguistic representations across multilingual Lexical Functional Grammar (LFG) grammars. A particular issue is that the grammars in the ParGram project are developed at different international sites. The approach that was established over several years relies on (i) a grammar code reviewing committee in which extensions to the existing representations are critically discussed, (ii) a technical tool for checking adherence to the best-practice feature declaration for linguistic representations, and (iii) a coordinated, systematic use of templates for expressing generalizations across lexicon entries and grammar rules. We compare the techniques used in practical LFG development with elements of Head-driven Phrase Structure Grammar (HPSG).

1 Introduction

At the very minimum, multilingual grammar development requires agreement on a common set of representations and broad agreement on the analyses of linguistic phenomena. Discussions about the proper analysis and representation of phenomena such as subject-verb agreement, case marking, relative clauses, or the treatment of adjectives or adverbials can be conducted at a very high linguistic level. Ultimately, though, differences in opinion are reified at a very low level of engineering. Since its inception, the Parallel Grammar (ParGram) project has therefore included a "feature committee," whose job is to find norms for the use and definition of a common multilingual feature space. Adherence to feature committee decisions is technically supported by a routine that checks the grammars for compatibility with a feature declaration. Parallelism at the level of grammatical constraints or descriptions is facilitated by systematic use of means of abstraction in the grammar specification code.

The ParGram project is an international collaboration aimed at producing broad-coverage computational grammars for a variety of languages ((Butt et al., 1999; Butt et al., 2002); see (Riezler et al., 2002) on the coverage of the English grammar). The grammars (to date of English, French, German, Japanese,[1] Norwegian, and Urdu) are written in the framework of Lexical Functional Grammar (LFG) (Kaplan and Bresnan, 1982; Dalrymple, 2001), and they are constructed using a common engineering and high-speed processing platform for LFG grammars: XLE (Maxwell and Ka-

[1]A Korean grammar is currently being ported from the Japanese grammar to determine how quickly a deep grammar can be developed when bootstrapped from the grammar of a typologically similar language (Kim et al., 2003).

plan, 1993).

In keeping with standard LFG practice, these grammars assign two levels of syntactic representation to the sentences of a language: a surface phrase structure tree (called a *constituent structure* or c-structure) and an underlying matrix of features and values (the *functional structure* or f-structure). The c-structure records the order of words in a sentence and their hierarchical grouping into phrases. The f-structure encodes the grammatical functions, syntactic features, and predicate-argument (dependency) relations conveyed by the sentence. F-structures are meant to encode a language universal level of analysis, allowing for crosslinguistic parallelism at this level of abstraction. For example, while the analysis of the English, French, and German versions of a sentence like (1) will necessarily differ at the c-structural level (different word orders, different numbers of auxiliaries), at the f-structure level all the grammars produce something like the dependency structure in Figure 1, in which the main predicate is the verb and the tense and aspect information is collected under the feature TNS-ASP (Butt et al., 1996). The language-particular dependencies between tense inflection, auxiliaries, and verbs are not encoded at f-structure.

(1) a. Tomorrow the letter will have arrived.

b. Demain la lettre sera
tomorrow the letter will-be

arrivée. (French)
arrived

c. Morgen wird der Brief
tomorrow will the letter

angekommen sein. (German)
arrive be

In this paper, we discuss some of the technical and organizational means that have been developed in the ParGram project to establish and enforce cross-grammar standards. Section 2 addresses the definition and comparison of allowable features and their proper values. Section 3 discusses the systematic use of templates in the grammar description language as a way of making generalizations explicit and transparent across grammars. Section 4 provides some discussion and a conclusion.

```
"Morgen wird der Brief angekommen sein."

  [PRED       'an#kommen<[41:Brief]>'
  [ADJUNCT   {  [PRED       'morgen']  }
  [          { 0[ADV-TYPE   temp     ]  }
  [            [PRED   'Brief'
  [            [NSEM   [COMMON count]
  [            [NTYPE  [GRAIN common]
  [SUBJ        [SPEC   [DET [PRED      'die']]
  [            [            [DET-TYPE  def  ]]
  [          41[CASE nom, GEND masc, INFL strong-det, NUM sg, PERS 3]
  [TNS-ASP     [ASPECT [FUT +_, PERF +_]
  [            [MOOD indicative, TENSE present]
  [TOPIC       [0:morgen]
21[CLAUSE-TYPE declarative, STMT-TYPE declarative, VTYPE main]
```

Figure 1: F-structure for (1c)

2 Defining the Feature Space

In this section, we discuss the feature space in the LFG ParGram grammars and how it is defined and regulated.

2.1 The Status of Feature Appropriateness Conditions

One way of ensuring parallelism in f-structure level analysis is to define a crosslinguistically relevant feature space in advance. The idea of theory-driven type/sort definitions for the feature structure representations as used in Head-driven Phrase Structure Grammar (HPSG; (Pollard and Sag, 1994)), for example, would appear to be ideal for such a purpose. The space of valid representations is restricted by the use of typed values for each feature, where the type of a feature-structure object defines exactly what embedded features are *appropriate*. Very elaborate type hierarchies have been proposed (see (Copestake, 2002) for implementation and documentation of typed feature structure grammars).

The grounding of the type hierarchy in the theoretical framework would seem to provide ideal support for a multilingual grammar writing effort (Bender et al., 2002). However, for use in broad-coverage grammars, the central role played by feature appropriateness conditions in HPSG theory can also pose problems. As the discussions in the HPSG community show, even quite fundamental configurations in the feature geometry have been subject to dispute and revision over the years.

The consideration of additional languages or phenomena may lead to insights that justify, or even require, potentially fundamental revisions to the feature space. This can pose considerable problems for a continuing grammar development effort. Moreover, it is not entirely clear whether there is (or should be) a crosslinguistically uniform type hierarchy, although this issue is being tested and explored in the Matrix project (Bender et al., 2002; Flickinger and Bender, 2003).

The methodology for ensuring parallelism and consistency in the ParGram project relies on checking feature appropriateness in a very similar way to HPSG (section 2.3); however, the conditions are established in a less theory-driven manner. The LFG formalism does not enforce feature appropriateness conditions *per se*. They are viewed as an engineering-level construct used for the purpose of consistency-checking. This encourages an explorative comparison of different possible representations: it is especially useful for phenomena not discussed in the theoretical literature. The resulting feature geometry is typically “leaner” than an HPSG feature geometry, which goes along with its lesser theoretical status and makes it less prone to major revision.

In the remainder of this section, we describe two tools used to ensure parallelism in the feature space: feature declarations and the feature table.

2.2 Feature Declarations

While c-structure analyses are subject to language particular variation by definition, the idea behind the f-structures is that they reflect a more language universal analysis. In order to maintain parallel grammar development, it is therefore vital to have identical features playing identical roles in all of the grammars.

For some features, the discussions and decisions to be made in a multilingual context are relatively straightforward. For example, the fact that there should be a CASE feature universally is generally undisputed, as well as the idea that the core values of such a feature should be at the very least NOM(inative), DAT(ive), and ACC(usative). However, one could question whether English does indeed need these features (e.g., (Hudson, 1995)) and whether they should be spelled the English way or, for example, the German way (e.g., AKK(usativ)). While these latter sorts of questions seem relatively low-level, the core of language individual analyses depends on the precise declaration and space of the features. For example, much discussion went into the definition of the tense/aspect features: deciding which attributes and values to use meant engaging in a discussion of the underlying theoretical treatment of tense/aspect.

In feature declarations, two basic types of feature values are distinguished: atomic (or constant) values and complex f-structure values.[2] For atomic values, the feature declaration specifies that the value of a feature FEAT (denoted by FEAT:$\rightarrow$; the right-arrow introduces the definition of the *possible values* of the feature to its left) has to be a member of a set of atomic values:

(2) FEAT:$\rightarrow \in$ {value1 value2 ...valueN}.

For complex f-structure values, the declaration specifies the features that the embedded feature structure may contain, using the subsumption operator $\ll$:

(3) FEATA:$\rightarrow \ll$ [FEATB1 ...FEATBN].

Consider the TNS-ASP feature and the features it calls in the feature declaration in (4). Values may be atomic, in which case a set of possible atoms is specified, as for MOOD, or they may be complex, in which case appropriate features for the embedded feature structure are specified, as for TNS-ASP. Note that the complex feature structure may consist of only a subset of these features.

(4) TNS-ASP: $\rightarrow \ll$ [MOOD PERF PROG TENSE].
MOOD: $\rightarrow \in$ {imperative indicative subjunctive}.
PERF:$\rightarrow \in$ {+_ −_}.
PROG: $\rightarrow \in$ {+_ −_}.
TENSE: $\rightarrow \in$ {fut null past pres}.

[2] A further type of feature values are closed-set values, as argued for in (Dalrymple and Kaplan, 2000). An indeterminate pronoun may, for example, have the case feature { nom, acc }. The feature declaration for a closed-set-typed case feature would involve using the set subsumption operator:
CASE: $\rightarrow \ll$ { nom gen dat acc }.

Just by examining the features in (4), it is possible to tell that tense-mood-aspect information is grouped together under the TNS-ASP feature (instead of having the individual features directly in the verb's f-structure). In addition, we can see that the grammars do not make use of composite tense/aspect values such as past perfect, pluperfect, or future perfect. Early on in the ParGram project, this type of composite feature value for tense/aspect was attempted for the German, English, and French grammars. However, not only did the cost of determining the right value turn out to be too great (since these tenses are often encoded as periphrastic constructions, a complex system of interdependencies needs to be checked), but it was difficult to establish a coherent crosslinguistic feature space. It was therefore decided to fall back on a more atomic encoding of tense/aspect, in which atomic features are simply registered under the TNS-ASP feature. That is, the English auxiliary *will* contributes the feature FUT, the English auxiliary *have* contributes the PERF +_,[3] etc. The more intricate problem of a precise and crosslinguistically valid semantic analysis of tense/aspect is left to a separate semantic component, which can base its analysis on the information collected under the TNS-ASP feature.

When the grammar is loaded, the feature declaration is checked against the compiled grammar rules. If any undefined features are found or if any undefined values of declared features are found, XLE returns an error message indicating the offending feature and where it appeared. The grammar cannot be loaded until the feature declaration violation is fixed.[4] For example, if the feature declaration allows for ADV-DEGREE `positive` and `comparative` but the grammar also has ADV-DEGREE `superlative`, e.g., as might occur when the grammar writer is adding superlative adverbs to the grammar, then XLE provides a message as in (5).

(5) Feature declaration violation near line 43, column 23 in file /pg/eng/standard/english-templates.lfg:
ADV-DEGREE cannot be equal to superlative

The XLE system allows for grammar adaptation whereby a standard grammar developed for a language can be modified to a specific corpus (Kaplan et al., 2002). Among the possible adaptations is to allow controlled changes of the feature declaration. In particular, the specialized grammar can use the standard features as a default, while overriding only those features which have changed in the specialized version. This way, any changes made to the feature declaration of the standard grammar are automatically incorporated into the specialized grammar. This is done in the grammar configuration file where the feature declaration is stated, as in (6) which gives priority to the EUREKA feature declaration while using the STANDARD one as a default.

(6) FEATURES STANDARD-ENGLISH EUREKA-ENGLISH.

The specialized feature table then only needs to contain newly introduced features, e.g., FIELD in (8a), or features whose values have changed, e.g., NSEM, in (8b).

(7) a. FIELD: $\rightarrow \in$ {cause item problem solution}.

b. NSEM: $\rightarrow \ll$ [PROPER TIME EUREKA-TYPE].

As we will see in the next section, the ability to include multiple feature declarations can also be used in a multilingual context to create common feature declaration which can then be specialized for particular languages.

[3]The use of the underscore in the value '+_' indicates its status as an *instantiated symbol*. This means that it is resource-sensitive in the sense that it can be introduced only once through a *defining* feature equation. In other places, only *constraining* equations can be used to check for the feature value. The most well-known instantiated symbols are the semantic forms used as values of the PRED feature. Here, instantiation excludes a duplication of identical material in various clause positions.

[4]In order to check for feature declaration violations in the lexicon, it is necessary to load the generator for the grammar since only in generation is the entire lexicon indexed. As with feature violations in the grammar, XLE will indicate the feature in question and where it was found; once the violations are fixed, the generator can be loaded.

Feature	ENGLISH	GERMAN	JAPANESE
CASE	acc gen nom	acc dat gen nom	acc gen nom
INF-FORM	bare to	zu	
TNS-ASP	MOOD PERF	FUT MOOD PASS-SEM	MOOD PROG
	PROG TENSE	PERF TENSE	TENSE
VTYPE	copular main modal	copular main modal	copular main

Table 1: Sample FEAT-TABLE

2.3 The Feature Committee and Feature Declaration Checking

Within ParGram, the definition of the multilingually relevant feature space is established via twice-yearly meetings among the grammar writers. Here, new developments in the individual grammars are checked and discussed. New features introduced for the grammar of a particular language are skeptically reviewed and are only sanctioned if they can be shown to have a universal application or if it becomes clear that the individual grammar could not have done without this feature. Every effort is made to keep the feature-space as small as possible and to assimilate new analyses within the existing feature-space. To facilitate this, the feature declarations are combined into a feature table (FEAT-TABLE) which has a column for each language and the features as rows; the cells are the values of the features for a particular language. A sample feature table is shown in Table 1.

The FEAT-TABLE can be used at a fairly high level to determine which grammars cover which contructions and what types of analyses they give to them. The FEAT-TABLE can also be used to determine trivial differences across the grammars, like the spelling of CASE values mentioned above.

One typical issue in which the FEAT-TABLE proved useful was the treatment of noun types. The precise classification of different types of nouns cannot be imported from theoretical linguistics directly into grammar development (unlike, for example, case or tense/aspect). Rather, as the grammars grew and had to parse different kinds of nouns (e.g., as encountered in the Wall Street Journal), individual grammar writers introduced more and more features to constrain the number and type of different analyses. These were all duly recorded in the FEAT-TABLE, enabling the grammar writers to easily compare their analyses and naming conventions and then agree on a standardized approach across the grammars. In particular, a very flat structure had arisen whereby most of the features were immediate attributes of the noun's predicate. Most of these were then rearranged under an NTYPE feature and further subdivided into syntactic (NSYN) and semantic (NSEM) features which in turn could have complex values. The syntactic values include information as to whether the noun is, for example, `common`, `count`, `gerund`, `partitive`, `pronoun`, or `proper`; the semantic values include information as to what type of proper noun it is (e.g. `location`, `name`). This more hierarchical structure allows the grammar writer to more easily envision the possible types of nouns in the grammar and to make reference to noun classes with fewer disjunctions.

The ability to invoke multiple feature tables in an override fashion was discussed in the previous section (Kaplan et al., 2002). In addition to providing a way of specifying multiple "dialects" of a given language/grammar, this idea can also be exploited in a multilingual setting. A "universal" or "common" feature declaration can be created for all languages. This common feature declaration is then invoked by all of the grammars. If a specific language needs additional features or feature values which are not in the common feature table, then these can be specified in the language particular feature table. This language specific table is then given higher priority than the common one, as in (8).

(8) FEATURES COMMON STANDARD-FRENCH.

If, for example, the English grammar needs to have an ACONSTR feature to control adjectives in specific constructions or needs to have a DEIXIS feature as a value of DET, then these features will appear in the English feature table as in (9).

(9) ACONSTR: $\rightarrow \in$ {equi extrapos obj-con raise}.
DET: $\rightarrow \ll$ [DEIXIS DET-TYPE PRED].

Once a language particular feature or feature value has been proposed, the feature committee can then examine the nonconforming specifications and decide whether the feature is actually something that is crosslinguistically relevant (but simply has not come up as yet as a part of the analyses in any of the other grammars) and therefore should be added to the common feature specification, whether the feature is the result of an aberrant analysis on the part of the grammar writer and therefore should be disallowed, or whether the feature indeed reflects a language particular characteristic and should therefore be included as part of the language specific feature declaration. The ability to invoke multiple feature declarations in an override fashion thus allows for more direct multilingual grammar development and highlights differences among the feature spaces of the languages.

3 Capturing Generalizations through Templates

The discussion in the previous section focused on the use of feature *representations* according to an agreed-upon definition of the feature space. This itself does not have necessary implications for the generality in which *descriptions of* or *constraints on these representations* are expressed in the grammars. It would be possible (although difficult) to write a grammar that does not capture any linguistic generalizations, but conforms to the feature declaration in each of its *ad hoc* rule statements.

Much of the grammar development work goes into the identification of suitable generalizations, since they are crucial for making the grammar readable, extensible, and maintainable. Although parallelism across multilingual grammars at the description level is not as indispensable as the representation-level parallelism (which may be sufficient for applications like machine translation), it is possible to avoid a significant amount of duplicate work by using a common specification of language-independent principles across grammars. Ultimately, a systematically organized grammatical description is also the best way to guarantee consistent representations.

An obvious example for a language-independent subsystem required for each of the grammars is the organization of the lexicon, particularly with regard to subcategorization. For example, verbs can universally be classified in terms of notions such as intransitive, transitive, or ditransitive. While languages may differ in terms of case marking or other requirements, there is a level at which this kind of generalization is very useful. When a new grammar is added to the project, the grammar writer can immediately expect to distinguish between intransitive, transitive, and ditransitive verbs. Given a universally defined lexical rule which generates passives from actives, the grammar writer may also expect to find passives in the language. The same goes for templates which define subject-verb agreement.

As was the case for the discussion of the feature space definition in section 2, one possible view to take with respect to the organization of the lexicon is that the linguistic theory should enforce a highly systematic structure. In HPSG, the feature structure entities representing (certain aspects of) words and phrases are organized in an inheritance hierarchy of types/sorts. This makes it possible for general properties holding for a large class of items to be specified once at a high level of abstraction and for the specific instantiations to inherit these properties.

LFG does not assume a sort hierarchy as part of the linguistic theory. However, means of abstraction like templates in the grammar specification make it possible to organize the lexicon in the same general way, following a specialization hierarchy where appropriate: more specific templates can call more general templates as part of their definition. The verbal subcategorization frame templates within most of the ParGram grammars provide a good example.

The central contribution of a verb subcategorization frame is a specification of the verbal stem and the number and type of arguments this verb subcategorizes for. The information about the stem is passed in as a variable, as shown in (10a). However, it is usually the case that other restrictions also apply to verbs with that basic subcategorization frame. These restrictions can be invoked by calling[5] other templates, as in (10b) which invokes templates that block particles and sentential subjects and that require a finite COMP. In turn, this template may be called by other templates that further restrict it, as in (10c) which requires a *that* complement and (10d) which requires a *whether* complement.

(10) a. V-SUBJ-COMP_PRED(_stem) =
(↑PRED)='_stem<(↑SUBJ)(↑COMP)>'.

b. V-SUBJ-COMP-FIN(_stem) =
@(V-SUBJ-COMP_PRED _stem)
@COMP-FIN @NO-PRT @NO-CL-SUBJ.

c. V-SUBJ-COMP-THAT(_stem) =
@(V-SUBJ-COMP-FIN _stem)
@(COMP-FORM that).

d. V-SUBJ-COMP-INT(_stem) =
@(V-SUBJ-COMP-FIN _stem)
@(COMP-FORM whether).

The example in (10) results in a partial template hierarchy as in (11).

(11)
V-SUBJ-COMP_PRED
V-SUBJ-COMP-FIN ...
V-SUBJ-COMP-THAT V-SUBJ-COMP-INT

Similar dependencies among the templates can be found throughout the template system, e.g., the nominal templates also form a loose hierarchy.

As before, the LFG grammar development philosophy allows the grammar writer of an individual language a large degree of freedom as to whether or not a highly structured subsystem of recursive template calls is used in a specific situation. This has the advantage that less studied linguistic phenomena can be added to the grammar without a major modeling effort; also, ill-understood exceptions to a given generalization can be included in a grammar without having to change the system of representations itself, something which would be required in a strictly inheritance-based framework. At the same time, this large degree of liberty requires a certain discipline in the context of a multi-site multi-language grammar development project. The code reviewing instance of the feature committee has proven to be one helpful way of controlling the freedom offered by the LFG formalism.

4 Discussion and Conclusions

The ParGram project attempts to test LFG for its universality and coverage and to see how far parallelism can be maintained across languages; previous ParGram work (and much theoretical analysis) has largely confirmed the universality claims of LFG theory. The f-structures produced by the grammars for similar constructions in each language have the same major functions and features, with minor variations across languages (e.g., the f-structures for French nouns have a gender feature but not the English f-structures). This uniformity has the computational advantage that the grammars can be used in similar applications and that machine translation (Frank, 1999) can be simplified. It also has the advantage that when new grammars (such as Urdu or Japanese) are added to the project, they can be bootstrapped relatively efficiently using the existing declared feature space and the existing templates.

The ParGram methodology of establishing consistency and parallelism of representations relies on (i) a feature committee in which extensions to the existing representations are critically reviewed and (ii) technical tools for checking adherence to a feature declaration for linguistic representations. The resulting system of checking feature appropriateness resembles the theory-driven typed feature structure signatures of HPSG; however, the feature appropriateness conditions are not predominantly theory-driven, but are established by best practice. This takes into account the specific needs of multi-site development, and seems to be more appropriate for broad-coverage grammars in which representations often have to be defined for phenomena not yet analyzed in the literature. A similar approach is taken in the systematic structuring of the grammatical descriptions: means of abstraction like templates can be used to implement a hier-

[5] Template calls are indicated by an @ before the name of template in question; templates can take arguments, as in @(COMP-FORM that), or not, as in @NO-PRT.

archical organization of linguistic generalizations where appropriate (but they can be side-stepped if necessary). Such a structured grammatical subsystem can be used across grammars in a multilingual context.

References

Emily Bender, Dan Flickinger, and Stephan Oepen. 2002. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the Workshop on Grammar Engineering and Evaluation*, pages 8–14. COLING 2002 workshop.

Miriam Butt, Maria-Eugenia Niño, and Frédérique Segond. 1996. Multilingual processing of auxiliaries in LFG. In D. Gibbon, editor, *Natural Language Processing and Speech Technology: Results of the 3rd KONVENS Conference*, pages 111–122.

Miriam Butt, Tracy Holloway King, Maria-Eugenia Niño, and Frédérique Segond. 1999. *A Grammar Writer's Cookbook*. CSLI Publications.

Miriam Butt, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. The Parallel Grammar project. In *COLING 2002: Workshop on Grammar Engineering and Evaluation*, pages 1–7.

Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Publications, Stanford, California.

Mary Dalrymple and Ronald M. Kaplan. 2000. Feature indeterminacy and feature resolution. *Language*, 76(4):759–798.

Mary Dalrymple. 2001. *Lexical Functional Grammar*. Academic Press, New York. Syntax and Semantics, volume 34.

Dan Flickinger and Emily Bender. 2003. Compositional semantics in a multilingual grammar resource. In *ESSLLI 2003 Workshop on Ideas and Strategies for Multilingual Grammar Development*. This volume.

Anette Frank. 1999. From parallel grammar development towards machine translation. In *Proceedings of MT Summit VII*, pages 134–142.

Richard Hudson. 1995. Does English really have case? *Journal of Linguistics*, 31:375–392.

Ronald M. Kaplan and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281. The MIT Press.

Ronald M. Kaplan, Tracy Holloway King, and John T. Maxwell III. 2002. The Parallel Grammar project. In *COLING 2002: Workshop on Grammar Engineering and Evaluation*, pages 29–35.

Roger Kim, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, Hiroshi Masuichi, and Tomoko Ohkuma. 2003. Multilingual grammar development via grammar porting. In *ESSLLI 2003 Workshop on Ideas and Strategies for Multilingual Grammar Development*. This volume.

John T. Maxwell, III and Ronald M. Kaplan. 1993. The interface between phrasal and functional constraints. *Computational Linguistics*, 19:571–589.

Carl Pollard and Ivan Sag. 1994. *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics. University of Chicago Press.

Stefan Riezler, Tracy Holloway King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell III, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of the ACL*.

Treebank-Based Multilingual Unification-Grammar Development

Aoife Cahill*, **Martin Forst**†, **Mairead McCarthy***, **Ruth O'Donovan***,
Christian Rohrer†, **Josef van Genabith***, **Andy Way***

*School of Computer Applications
Dublin City University
Dublin 9, Ireland

†Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
D–70174 Stuttgart, Germany

Abstract

Broad-coverage, deep unification grammar development is time-consuming and costly. This problem can be exacerbated in multilingual grammar development scenarios. Recently (Cahill *et al.*, 2002) presented a treebank-based methodology to semi-automatically create broad-coverage, deep, unification grammar resources for English. In this paper we present a project which adapts this model to a multilingual grammar development scenario to obtain robust, wide-coverage, probabilistic Lexical-Functional Grammars (LFGs) for English and German via automatic f-structure annotation algorithms based on the Penn-II and TIGER treebanks. We outline our method used to extract a probabilistic LFG from the TIGER treebank and report on the quality of the f-structures produced. We achieve an f-score of 66.23 on the evaluation of 100 random sentences against a manually constructed gold standard.

1 Introduction

Parsing is an important step in natural language processing as syntactic structure is a prime determinant for semantic interpretation in the form of predicate-argument structure, deep dependency structure or logical form. Rich unification (or rather: constraint) grammars such as LFG or HPSG model both (morpho-) syntactic and semantic information.

Manually scaling deep unification grammars to real text (such as the Penn-II treebank, Marcus *et al.*, 1994), however, is extremely time-consuming, costly and requires considerable linguistic and processing expertise, as it involves person-years of concerted grammar, lexicon and system (processing platform) development effort. What is more, few hand-crafted grammars achieve full coverage of the target corpus. Indeed, the only hand-crafted, deep unification grammar scaled to the full Penn-II treebank we are aware of is the English LFG grammar developed as part of the ParGram project at Xerox PARC (Riezler *et al.*, 2002; Butt *et al.*, 2002). This situation, we suspect, is even worse for languages other than English, as they have received considerably less (linguistic and computational linguistic) attention. Accordingly resource problems can be exacerbated in multilingual grammar development scenarios.[1]

Recently (Cahill *et al.*, 2002) presented a treebank-based methodology to semi-automatically create wide-coverage, deep, unification LFG grammar resources for English. The method is based on an automatic annotation algorithm that annotates Penn-II treebank trees with LFG f-structure information. F-Structures are recursive attribute-value structures approximating to predicate-argument-modifier representations. Alternatively, f-structures can be viewed as encodings of deep dependency re-

[1]Sometimes multilingual grammar development offers unique advantages unavailable to monolingual grammar development: these are cases where an existing grammar for language A can be migrated to a closely related language B for which such resources do not exist and thus significantly boost grammar development for language B. (Gamon *et al.*, 1997)

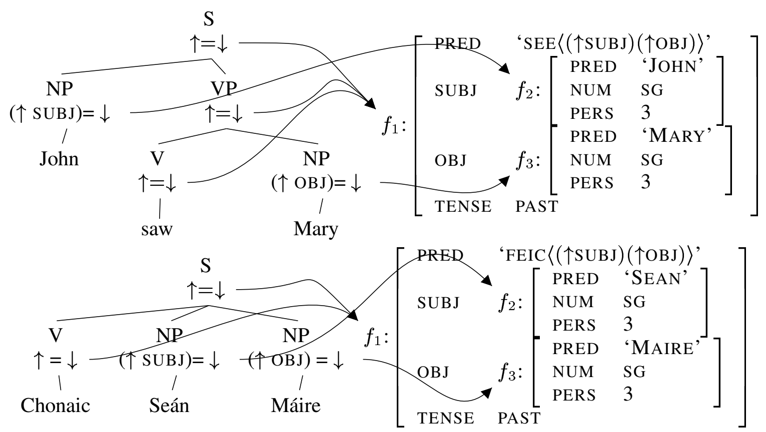


Figure 1: C- and f-structures for an English and corresponding Irish sentence

lations. (Cahill *et al.*, 2002) show how based on the f-structure annotated version of Penn-II, wide-coverage, robust, probabilistic LFG grammars can be derived to parse Penn-II data.

In this paper we first introduce LFG and, in particular, the level of f-structure representation and motivate why we think LFG provides a suitable representation format for multilingual grammar development. Next we present the basic ideas underlying the approach presented in (Cahill *et al.*, 2002) as applied to English. We outline how this approach can be adapted and migrated to a different language and treebank resource, namely German and the TIGER treebank (Brants et. al, 2002). German is substantially less configurational than English, and the TIGER treebank annotation consists of graphs with crossing edges rather than trees with traces (as in Penn-II). In addition, the TIGER treebank features considerably richer functional annotations than those provided in the Penn-II resource. We outline how LFG grammars for German can be derived from the f-annotated TIGER resource. We extract a probabilistic LFG and run some parsing experiments. We evaluate both the quality of the automatic annotation of the treebank, and the output of the parser which uses the probabilistic LFG. Finally, we present current work on automatic extraction of lexical resources from the f-structure annotated Penn-II treebank and recent work on long-distance dependencies. We then conclude and outline further work.

2 Lexical-Functional Grammar

Lexical-Functional Grammar (Kaplan and Bresnan, 1982; Bresnan, 2001) is an early member of the family of unification (or constraint) grammars (such as FUG, PATR-II, GPSG, CUP or HPSG). Minimally, LFG involves two levels of representation: c(onstituent)-structure and f(unctional)-structure. C-structure captures language specific phenomena such as word order and the grouping of constituents into larger phrases in the form of context-free trees. F-structure represents abstract syntactic functions such as subj(ect), obj(ect), pred(icate) etc. in the form of recursive attribute-value structures. The basic idea is that while languages may differ markedly with respect to surface realisation (c-structure), they may still exhibit very similar abstract syntactic functional representations (f-structure). Figure 1 illustrates this point. Irish is typologically a VSO-language, while English is an SVO-language. The same proposition expressed in Irish and English exhibits markedly different c-structure configurations but is associated with isomorphic (up to the values of PRED nodes) f-structure representations.

C-structure and f-structure representations are related in terms of "functional annotations" of the form $\uparrow \ldots =\downarrow \ldots$ to tree nodes, i.e. attribute-value structure equations (or more generally: disjunctive, implicational and negative constraints) describing f-structures.

F-structures approximate to predicate-argument-modifier representations, simple logical forms (van Genabith and Crouch, 1996) or deep dependency relations. LFG is particularly attractive for multilingual grammar development as the level of f-structure representation abstracts away from language-specific surface realisation (Butt *et al.*, 1999). At the same time LFG provides a precise, flexible, computationally tractable and non-transformational interface between c-structure and f-structure representation for both parsing and generation (Butt *et al.*, 2002). Unlike other unification grammar formalisms, LFG has enjoyed a substantial body of work on automatic f-structure annotation architectures summarised in (Cahill *et al.*, 2002; Frank *et al.*, 2003). These approaches automatically annotate (treebank or parse-generated) trees with f-structure equations to generate f-structures for those trees.

3 F-structures from Penn-II

(Cahill *et al.*, 2002) present an automatic f-structure annotation algorithm for the trees in the Penn-II treebank. They show how wide-coverage, robust probabilistic LFGs can be derived automatically. Given a tree, the task of f-structure annotation is to annotate tree nodes automatically with f-equations. As a simple example, consider the following CFG rule (i.e. a local tree of depth 1):

NP → DT ADJP NN SBAR

Such a configuration would be associated with f-structure annotations as follows:

NP →	DT	ADJP	NN	SBAR
	$\uparrow$ SPEC = $\downarrow$	$\downarrow\in\uparrow$ ADJ	$\uparrow=\downarrow$	$\downarrow\in\uparrow$ RELMOD

This indicates that the NN is the head of the NP, the DT is a specifier, the adjective phrase is part of the modifying adjunct set, and the SBAR is a member of the RELMOD set.

The annotation algorithm automatically transforms trees into head-lexicalised trees using a variant of (Magerman, 1994)'s head rules and then uses configurational, categorial, functional tag and trace information encoded in the Penn-II treebank trees to associate tree nodes with f-structure equations from which a constraint solver generates f-structures. The annotation is evaluated quantitatively (Table 1) for the 48,424 treebank trees (without FRAG or X constituents) and qualitatively (Table 2)

# f-str. frags	# sent	percent
0	120	0.25
1	48304	99.75

Table 1: Coverage & fragmentation results

	all annotations	preds-only
Precision	0.93	0.94
Recall	0.90	0.87

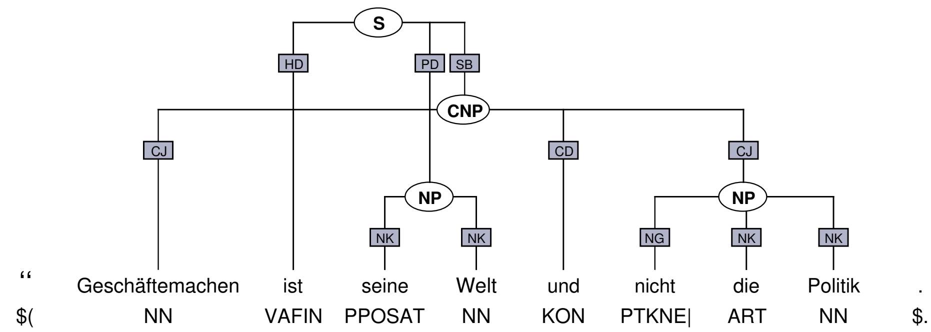
Table 2: Precision and Recall on f-structures against a manually encoded set of 105 gold standard f-structures from section 23

Based on this resource (Cahill *et al.*, 2002) derive two parsing architectures to parse new text: a *pipeline* and an *integrated* model. In the pipeline model, a PCFG is extracted from the unannotated treebank and used to parse new text. The resulting context-free trees are then passed into the automatic annotation algorithm to generate f-structures. In the integrated model an annotated PCFG is extracted from the f-structure-annotated Penn-II resource and used to parse new text. This results in annotated parse trees, from which an f-structure can be generated. The parsers are trained on sections 02-21 and evaluated against section 23. The grammars are both robust and wide-coverage. They achieve 81.2% f-score against the approx. 2,400 trees (section 23) and 60.6% f-score against the gold-standard f-structures. More recent experiments, with an improved annotation algorithm, yield an f-score of over 75% against the gold-standard f-structures.

4 From TIGER to a German LFG

The TIGER treebank (Brants et. al, 2002) is a corpus of (currently) 36,000 syntactically annotated German newspaper sentences. The annotation consists of generalised graphs, which may contain crossing and secondary edges. Edges are labelled, so that a TIGER tree encodes both phrase-structural information and dependency relations.

(Forst, 2003) converts the TIGER graphs directly into f-structures. However, in order to be able to extract an annotated PCFG which can be used to parse text into f-structures, we require trees that have been



Business is his world and not the politics.
“Business is his world, not politics.”

Figure 2: TIGER graph #45, containing crossing edges

annotated with f-structure equations, rather than the f-structures themselves.

Since the structure of the TIGER corpus is quite different to that of the Penn-II Treebank, the approach taken to annotating the TIGER corpus with f-structure equations differs from the approach described earlier. German does not usually rely on positional information to express functional information, a feature of English that was heavily exploited previously. However, the TIGER corpus aims to provide functional information by way of labelled edges in the graphs. By exploiting these labels we can annotate the TIGER corpus with f-structure equations.

4.1 From Graphs to Trees

The first stage in annotating the TIGER corpus with f-structure equations is to convert the TIGER graphs into trees similar to those found in the Penn-II Treebank. Traces are used to represent the crossing edges. Secondary edges have not been incorporated into the annotation procedure at this stage.[2] Although these edges obviously contain vital information for the generation of f-structures, this information is currently not utilised, since their treatment is unclear. However, we hope to exploit them in future work.

Figures 2 and 3 illustrate how traces are used to represent crossing edges. The TIGER graph indicates by means of crossing edges the fact that both

```
(
  ($*LRB* ``)
  (S
    (CNP-SB
      (NN-CJ Geschäftemachen)
      (*T1*-CD -)
      (*T2*-CJ -)
    )
    (VAFIN-HD ist)
    (NP-PD
      (PPOSAT-NK seine)
      (NN-NK Welt)
    )
    (KON-*T1* und)
    (NP-*T2*
      (PTKNEG-NG nicht)
      (ART-NK die)
      (NN-NK Politik)
    )
  )
  ($. .)
)
```

“Geschäftemachen ist seine Welt und nicht die Politik.”

Figure 3: TIGER graph #45 transformed into a Penn-II style tree with indexation

Geschäftemachen and *und nicht die Politik* form a discontinuous constituent, in the middle of which the rest of the sentence appears. [3]

4.2 Annotation of Derived Trees

The annotation of the trees which result from the above transformation is a two-stage process, with

[2]Secondary edges contain information relating to re-entrancies (e.g. a shared subject in a coordination construction).

[3]Thanks to Michael Schiehlen who provided the code to convert the graphs into corresponding trees with traces

```
(TOP
  ($*LRB* ``)
  (S[up=down]
    (CNP-SB[up-subj=down]
      (NN-CJ[down-elem=up:conj]
                     Geschäftemachen)
      (*T1*-CD -)
      (*T2*-CJ -)
    )
    (VAFIN-HD[up=down] ist)
    (NP-PD[up-xcomp_pred=down]
      (PPOSAT-NK[up-spec:poss=down] seine)
      (NN-NK[up=down] Welt)
    )
    (KON-*T1*[up:subj=down] und)
    (NP-*T2*[down-elem=up:subj:conj]
      (PTKNEG-NG[down-elem=up:adjunct]
                          nicht)
      (ART-NK[up-spec:det=down] die)
      (NN-NK[up=down] Politik)
    )
  )
  ($. .)
)
```

Figure 4: The tree in Figure 3 after automatic annotation

both pre- and post-editing of the annotated trees.

The preprocessing is a simple walk through the tree in order to build a lookup table for the trace nodes. This is needed since often the trace occurs before the actual node in the tree and the information on the actual node is needed in order to assign an f-structure equation to the trace node.

The first stage attempts to assign an f-structure equation to each node based on the functional labels in the tree. We have compiled a lookup table with default f-structure equations for each functional label. E.g., the default entry for the SB (subject) label is $\uparrow$ SUBJ = $\downarrow$. It is also possible to overwrite the default entries. E.g. the NK label (noun kernel element) alone is often ambiguous, though given some context, it is often straightforward to determine the f-structure equation required. For example an ART (article) node with an NK label can usually be annotated $\uparrow$ SPEC:DET = $\downarrow$.

The second stage in the annotation process involves overwriting the default annotations in certain situations. These include:

- Determining the object of pre- and post-positions, labelled AC (adpositional case marker);
- Determining the behaviour of the CP (complementiser) labelled node; [4]
- Determining the head of a coordination phrase with more than one coordinating conjunction.

Figure 5 illustrates how the flat analysis of a German PP can be annotated to give the correct f-structure analysis.

Finally, a post-processing stage explicitly links trace nodes and the reference node. This involves adding equations such as $\uparrow$XCOMP:OBJ = $\downarrow$ to nodes with trace information.

Figures 3 and 4 illustrate a complete annotation of a TIGER tree and Figure 6 illustrates the resulting f-structure.

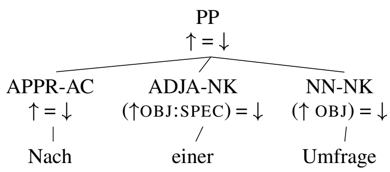


Figure 5: A flat analysis of a German PP and its f-structure annotations

4.3 Extracting an LFG from TIGER

Using the above annotation method we automatically annotate the TIGER corpus with f-structure equations. We can then quite simply read off a CFG from this annotated corpus, resulting in an *annotated* PCFG for German. We then use a standard parser to parse with this grammar, using Viterbi pruning to always obtain only the most probable parse. Using the same method as described in (Cahill *et al.*, 2002) we collect the f-structure annotations from the resulting parse tree and use a constraint solver to produce an f-structure.

5 Evaluation

In order to evaluate the quality of the grammar extracted, we set sentences 8001-10000 aside for

[4]In our analysis, true complementisers, i.e. *daß* and *ob*, only contribute a COMP-FORM feature to the f-structure, whereas other conjunctions contribute a semantic form that governs an object.

```
subj : conj : 1 : pred : 'Geschäftemachen'
              2 : spec : det : pred : die
                  adjunct : 3 : pred : nicht
                  pred : 'Politik'
       coord_form : und
xcomp_pred : spec : poss : pred : pro
             pred : 'Welt'
pred : ist
```

Figure 6: The f-structure produced as a result of automatically annotating the tree in Figure 3

# f-str. frags	# sent	percent
0	153	0.42
1	35191	96.46
2	1054	2.89
3	77	0.21
4	2	0.006
5	1	0.003
6	1	0.003
7	3	0.008

Table 3: Coverage & fragmentation results of German Annotation Algorithm

	Preds Only Evaluation
Precision	86.79
Recall	84.71
F-Score	85.74

Table 4: Evaluation of the f-structures produced by automatically annotating the TIGER trees

testing purposes and developed a gold standard of 100 sentences extracted randomly from these sentences. The 100 sentences were first converted to f-structures using the methodology outlined in (Forst, 2003). They were then converted into a set of dependency relations, similar to those of the PARC Dependency Bank (King *et al.*, 2003), which were then checked and corrected manually. Using this gold-standard it was possible to use various evaluation metrics to evaluate the grammar.

5.1 Evaluation of the Automatic Annotation Algorithm

We first established the coverage of the annotation algorithm on the entire TIGER corpus. Table 3 illustrates the results. Ideally we would like to generate just one f-structure per sentence. There are however, a number of sentences that receive more than one f-structure fragment. This is mainly due to sentences such as *Bonn, 7. September*, where there is no clear relation between the elements of the "sentence" and where we do not wish to enforce a relation for the sake of having fewer fragments. We believe that these "sentences" are in fact fragments and should be treated accordingly. There are also a small number of sentences which do not receive any f-structure. This is as a result of feature clashes in the annotated trees, most if which are caused by annotation discrepancies. We also evaluate the quality of the annotation against our manually constructed gold-standard of 100 sentences. Table 4 illustrates that currently our automatic annotation receives an f-score of 85.74% when we compare the dependency relations generated by automatically annotating the 100 TIGER trees to the gold-standard relations. We expect this figure to improve as we refine the algorithm.

5.2 Evaluation of the Grammar

An annotated grammar was extracted from the TIGER corpus (excluding the 2000 sentences set aside for testing). Using this grammar, the 2000 test sentences were parsed using Helmut Schmid's BitPar parser (p.c.), which always produced the most probable analysis. 1992 sentences received a parse, from which 1974 received at least one f-structure fragment. Only 3.3% of the sentences received a fragmented f-structure, mostly due to the nature of these sentences. Table 5 contains the entire fragmentation percentages of the f-structures generated by the grammar. Table 6 illustrates the results of evaluating the trees produced by the parser against the same trees as produced in the conversion from TIGER trees to Penn-II style trees. These annotated trees were then processed into f-structures. To evaluate the quality of the f-structures produced by the parser, we again evaluated the 100 sentences selected randomly from the 2000 test sentences. Table 7 illustrates the results.

# f-str. frags	# sent	percent
0	26	1.3
1	1908	95.4
2	61	3.05
3	5	0.25

Table 5: Coverage & fragmentation results of parsing with the annotated grammar

	Unlabelled	Labelled
Precision	66.05	63.22
Recall	69.29	66.33
F-Score	67.63	64.74

Table 6: Evaluation of the trees produced by the parser

	Preds Only Evaluation
Precision	70.17
Recall	62.70
F-Score	66.23

Table 7: Evaluation of the f-structures produced by the parser

6 Lexical Resources and LDDs

In ongoing work we have extracted lexical resources such as subcat frames (LFG semantic forms) and long-distance dependency paths from the f-structures generated from the Penn-II treebank. (Cahill *et al.*, 2003b) We apply these resources during parsing to resolve long-distance dependencies in f-structure. (Cahill *et al.*, 2003a). We expect to be able to extract similar resources from the f-structure annotated TIGER bank.

7 Conclusions

We have outlined what we believe is a novel, semi-automatic approach to multilingual grammar development based on treebank resources and automatic f-structure annotation algorithms. This method can offer substantially reduced grammar development cost if a treebank is available. Depending on the size of the treebank the method can deliver robust and wide-coverage unification grammars. The method has been developed and tested for English (Cahill *et al.*, 2002). We have illustrated how this method can be adapted to German and the TIGER treebank resource. We present results of parsing 2000 sentences with a probabilistic LFG extracted automatically from the f-structure annotated TIGER treebank. The parser produces f-structures which receive an f-score of 66.23 when evaluated against our gold-standard. We have argued for LFG as a suitable framework for multilingual grammar development. However, nothing in the methodology presented here precludes their application to other languages or corpora. In addition, different annotation schemes could also be applied to automatically derive other representations, e.g. HPSG attribute-value structures, dependency structures etc. Finally, we do not believe that semi-automatic, treebank-based, multilingual grammar development necessarily competes with traditional, predominantly manual grammar development: indeed, flexible integration of hand-crafted, deep, HPSG grammars and automatically extracted, pure CFG-based topological treebank grammars (not unification grammars) has recently been demonstrated in (Crysman *et al.*, 2002; Frank *et al.*, 2003b) and we consider this to be a promising direction for future research in both mono- and multilingual wide-coverage, deep unification grammar development.

References

Becker, M. and A. Frank 2002. ‘A Stochastic Topological Parser for German’. *Proceedings of COLING 2002*, Taipei, Taiwan.

Brants, T., S. Dipper, S. Hansen, W. Lezius and G. Smith 2002. The TIGER Treebank. In: Hinrichs and Simov (eds.), *Proceedings of the first Workshop on Treebanks and Linguistic Theories (TLT’02)*, Sozopol, Bulgaria

Brants, T. and O. Plaehn 2000. ‘Interactive Corpus Annotation.’ In *Second International Conference on Language Resources and Evaluation (LREC-2000)*, Las Palmas, the Canary Islands.

Bresnan, J. 2001. *Lexical-Functional Syntax*. Blackwell, Oxford.

Butt, M., T.H. King, M.E. Niño and F. Segond, 1999. *A grammar writer’s cookbook*. Stanford, CA. CSLI Publications

Butt, M., H. Dyvik, T.H. King, H. Masuichi and C. Rohrer, 2002. ‘The parallel grammar project.’ *Proceedings of COLING 2002, Workshop on Grammar Engineering and Evaluation* pp. 1-7.

Cahill, A., M. McCarthy, J. van Genabith and A. Way; 2002. ‘Parsing with PCFGs and Automatic F-Structure Annotation.’ In M. Butt and T. Holloway-King (eds.): *Proceedings of the Seventh International Conference on LFG* CSLI Publications, Stanford, CA., pp.76–95.

Cahill, A., M. McCarthy, R. O’ Donovan, J. van Genabith and A. Way; 2003. ‘Lexicalisation of Long-Distance Dependencies in a Treebank-Based, Statistical LFG Grammar.’ In: *Proceedings of the Eighth*

International Conference on LFG Albany, NY. (to appear)

Cahill, A., M. McCarthy, R. O' Donovan, J. van Genabith and A. Way; 2003. 'Extracting Large-Scale Lexical Resources for LFG from the Penn-II Treebank.' In: *Proceedings of the Eighth International Conference on LFG* Albany, NY. (to appear)

Crysmann, B., A. Frank, B. Kiefer, St. Müller, G. Neumann, J. Piskorski, U. Schäfer, M. Siegel, H. Uszkoreit, F. Xu, M. Becker and H. Krieger 2002. 'An Integrated Architecture for Deep and Shallow Processing' Proceedings of the *40th Annual Meeting of the Association for Computational Linguistics (ACL'02)* Philadelphia, PA.

Forst, M. 2003. 'Treebank Conversion - establishing a test suite for a broad-coverage LFG from the TIGER treebank' In *Proceedings of the EACL Workshop on Linguistically Interpreted Corpora (LINC'03)*, Budapest, Hungary

Frank, A. 2000. 'Automatic F-Structure Annotation of Treebank Trees.' In: (eds.) M. Butt and T. H. King, *The fifth International Conference on Lexical-Functional Grammar*, The University of California at Berkeley, 19 July - 20 July 2000, CSLI Publications, Stanford, CA.

Frank, A., L. Sadler, J. van Genabith and A. Way 2003. 'From Treebank Resources to LFG F-Structures.' In: (ed.) Anne Abeille, *Treebanks: Building and Using Syntactically Annotated Corpora*, Kluwer Academic Publishers, Dordrecht/Boston/London, to appear (2003)

Frank, A., M.Becker, B.Crysman, U.Kiefer and U.Schaefer 2003. 'TopP meets HPSG' Proceedings of the *41st Annual Meeting of the Association for Computational Linguistics (ACL'03)* Sapporo, Japan. (to appear)

Gamon, M., C. Lozano, J. Pinkham and T. Reutter 1997. 'Practical Experience with Grammar Sharing in Multilingual NLP.' In J. Burstein and C. Leacock (eds.): *Proceedings of the workshop From Research to Commercial Applications: Making NLP Work in Practice*, at the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the Association for Computational Linguistics (ACL-EACL'97), July 11-12, 1997. Madrid, Spain. Madrid: UNED, 49-56.

Kaplan, R. and J. Bresnan 1982. 'Lexical-functional grammar: a formal system for grammatical representation.' In Bresnan, J. (ed.) *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge Mass. 173–281.

King, T.H., R. Crouch, S. Riezler, M. Dalrymple, and R.M. Kaplan 2003. 'The PARC 700 Dependency Bank.' In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora*, EACL'03, Budapest.

Magerman, D. 1994. *Natural Language Parsing as Statistical Pattern Recognition*, PhD Thesis, Stanford University, CA.

Marcus, M., G. Kim, M. A. Marcinkiewicz, R. MacIntyre, M. Ferguson, K. Katz and B. Schasberger 1994. 'The Penn Treebank: Annotating Predicate Argument Structure.' In: *Proceedings of the ARPA Human Language Technology Workshop*. Princeton, NJ.

Riezler, S., T.H. King, R. Kaplan, R. Crouch, J.T. Maxwell and M. Johnson. 2002. 'Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques' Proceedings of the *40th Annual Meeting of the Association for Computational Linguistics (ACL'02)* Philadelphia, PA.

van Genabith, J. and R. Crouch 1996. 'Direct and Underspecified Interpretations of LFG f-Structures.' In: *COLING 96*, Copenhagen, Denmark, Proceedings of the Conference. 262–267.

Controlling spurious ambiguity with extraposition

Berthold Crysmann
DFKI GmbH
Saarbrücken, Germany
crysmann@dfki.de

Abstract

In this paper, I shall propose an implementation of relative clause extraposition in German. The proposal builds on (Kiss, 2002) who treats relative clause extraposition as an anaphoric process by means of percolation of anchors to which the relative clause is bound. I discuss several sources of spurious ambiguity in Kiss's original formulation and suggest a two-step percolation of anchors that crucially distinguishes right-peripheral from central or left-peripheral percolation. Since extraposition is fairly productive, and phrase-structure alternates between head initial (prepositional phrases, V-initial) and head-final structures (postpositional phrases, V-final), German provides a good testing ground for techniques controlling spurious ambiguity that may easily be ported to languages where phrase structure is more canonical and/or extraposition more restricted.

The phenomenon of extraposition in German has been an area of central concern in the context of theoretical approaches to an HPSG of German. Although there is now overwhelming consent (Nerbonne, 1994; Kathol, 1995; Müller, 1999; Kathol, 2000) that extraposion of relative clauses is best regarded as a word order phenomenon, these approaches all rely on the concept of word order domains, a mechanism that is so far absent from efficient implementations of the HPSG formalism, e.g. the runtime system PET (Callmeier, 2000) and the development platform LKB (Copestake, 2001), which both assume continuous constituents only.[1]

A viable, though highly inefficient solution would be to simulate domain-based extraposition by way of rightward extraction: as relative clauses are always modifiers, and therefore optional, such a move will inevitably introduce a fair amount of local ambiguity. Since extraposition is not strictly local (Müller, 2001; Kiss, 2002), but only upward bounded, a trace needs to be hypothesised for every NP contained in a clause. Owing to the non-locality of the process, the ambiguity thus created can only be resolved at the clausal level.

1 RC extraposition without discontinuous constituents (Kiss, 2002)

An interesting theoretical alternative to linearisation-based extraposition has recently been proposed by (Kiss, 2002; Kiss, 2003). He argues that extraposition differs from extrac-

[1] An exception here is Stefan Müller's Babel system which permits direct processing of discontinuous constituents and for which also a large-scale grammar is available (Müller, 1996; Müller, 2002). W.r.t. performance, however, the Babel system cannot currently compete with systems such as PET in the context of real-world applications: Comparing the relative performance differences between PET and Babel, (Müller, 2002) reports that his PET grammar performed around 13 times faster than the Babel grammar, when parsing the Verbmobil corpus. Taking into account recent performance improvements of around factor 2–3 for the DFKI German HPSG grammar (Crysmann, 2003), we can expect an even higher factor.

tion proper in that it does not seem to obey island constraints. Instead he suggests to model extraposition as an anaphoric process where a relative clause retricts an index contributed by an antecedent contained in the clause. This basic idea is expressed by his principle of Generalised Modification:

(1) Generalized Modification:
The index of a modifying phrase has to be identified with a suitable index contained in the phrase to which the modifier is adjoined. (Kiss, 2002, p. 5)

Technically, this is achieved by means of percolating the HANDEL and INDEX of every NP in the clause via a set-valued nonlocal feature ANCHORS and requiring that the relative clause's INDEX and HANDEL features be token-identical with those of a member in ANCHORS.

(2) Anchor Projection Principle:
The anchors set of a headed phrase consists of the union of the anchors sets of the daughters less those anchors that are specified as TO-BIND|ANCHORS on the head-daughter. (Kiss, 2002, p. 29)

Clause-locality, i.e. upward boundedness of extraposition is modelled by imposing appropriate restrictions on phrasal and lexical signs capable of closing off a sentential projection. Thus, (Kiss, 2002) proposes that e.g. complementisers identify their TO-BIND|ANCHORS set with the ANCHORS set of their sentential complement, effectively blocking further percolation:

(3)
$$\begin{bmatrix} \text{HEAD} & \textit{comp} \\ \text{COMPS} & \left\langle \text{S}\begin{bmatrix}\text{ANCHORS} & \boxed{1}\end{bmatrix}\right\rangle \\ \text{TO-BIND}\,|\,\text{ANCHORS} & \boxed{1} \end{bmatrix}$$

A similar assumption is made for head-specifier structures, which he assumes license V2 structures.

Thus, sentences like (4) can effectively be ruled out, due to the lack of an appropriate anchor for the relative clause:

(4) * Ulrich hatte zugegeben, daß die Karte
Ulrich has admitted that the ticket
gestohlen war, als er getrunken
stolen was when he drunk
hatte, die er gefunden hatte.
has which he found had
(Kiss, 2002)

The proposal advanced by (Kiss, 2002) is quite powerful, in that it is capable of modelling relative clause extraposition not only at the sentential level (5), but also at the level of PPs and participial clauses (see (6)).

(5) Man hatte **den Boten** beschimpft,
one has the messenger insulted
der den Befehl überbrachte.
who the command delivered
'The messenger was insulted who delivered the command.' (Kiss, 2002)

(6) Man muß nicht gleich [**eines**
one must not immediately an
Sachverständigen wegen, **der** seinen
expert because who his
Mund nicht halten kann,] die gesamte
mouth not keep can the whole
Konkurrenz kopfscheu machen.
competitors nervous make
'It does not make sense to confuse one's competitors just because an expert could not keep quiet.' (Kiss, 2002, p. 21)

Furthermore, from a processing point of view, the anaphoric perspective on relative clauses is to be preferred over any movement approach, as indices can be added without creating any local ambiguity, and processing cost is thus delayed until the point where a relative clause is actually found and an index needs to be retrieved.

2 Controlling Spurious ambiguity

A drawback of (Kiss, 2002), however, is that this approach introduces a fair amount of spurious ambiguity, a problem already recognised, in part, by the author (p. 32). Spurious ambiguity arises from the fact that indices can, in principle, be retrieved at any point along the percolation path, giving additional extraposed analyses for modifiers that may equally be attached locally. Upon further scrutiny, we can actually distinguish two general cases here: one with head-final and one with head-initial structures.

(7) Sie hatte den Termin verschoben am
she had the appointment rescheduled on
Montag.
Monday
'She had rescheduled the appointment on Monday.' (Kiss, 2002)

Under (Kiss, 2002)'s approach, the above sentence receives two syntactic analyses, one with low attachment for the PP, corresponding to modification of the rescheduling event, and another one, with high attachment, capable of deriving modification of both the noun *Termin* and the verb *verschoben*.

A similar case of spurious ambiguity can be found with NPs. Here, low attachment (to N or N′) is possible, on a par with modification of the percolated anchor on NP. (Kiss, 2002) recognises the issue, but argues that both in-situ and extraposed modification must be permitted independently, and that the problem is counter-balanced by the empirical coverage of the approach.

Contrary to (Kiss, 2002), I claim that the spurious ambiguity problem is much more pervasive, necessitating a refinement of Kiss's original proposal.

(8) Man beschimpfte **den Boten**, **der**
one insulted the messenger who
den Befehl überbrachte.
the command delivered
'The messenger was insulted who delivered the command.'

(9) Man muß nicht gleich [wegen
one must not immediately because.of
eines Sachverständigen, **der** seinen
an expert who his
Mund nicht halten kann,] die gesamte
mouth not keep can the whole
Konkurrenz kopfscheu machen.
competitors nervous make
'It does not make sense to confuse one's competitors just because an expert could not keep quiet.'

The sentences in (8) and (9), which correspond quite closely to (5) and (6), respectively, illustrate that head-initial structures are another source for spurious ambiguity. Again, Generalised Modification permits restriction of the same index at several points along the percolation path.

If we depart from the assumption that in-situ modification must always be permitted, a solution to the problem will involve further specification of the conditions under which a percolated anchor can be accessed.

What we need to do is, first, draw a distinction between a local anchor and percolated anchors, and second, distinguish inherited anchors according to whether they are already available for modification or not. Availability of inherited anchors will then depend on the intervention of other syntactic material between the modifier and the semantic target of modification. To implement this, it will be necessary to divide ANCHORS into two lists[2], ACTIVE and INERT.

(10) $$\left[\textsc{synsem}\,|\,\textsc{nonloc}\,|\,\textsc{anc}\left[\begin{array}{ll}\textsc{active} & \textit{*diff-list*}\\ \textsc{inert} & \textit{*diff-list*}\end{array}\right]\right]$$

Given binary branching structures, spurious ambiguity of the kind described above can only arise in two situations: either with percolation along a head projection path or with percolation at the right edge. In order to control for the latter, all we need to do is to make composition of the ANCHORS lists sensitive to the position of the daughter from which they inherit: thus, the INERT value of the mother will correspond to the INERT value of the right daughter, whereas the ACTIVE value of the mother will be the concatenation of the ACTIVE values of the daughters with the INERT value of the left daughter. In other words, to make a percolated anchor available for modification by an extraposed phrase, it needs to be inherited, once, from a left daughter.

[2] Owing to the absence of sets in LKB and PET, we use difference lists to percolate bags of anchors.

$$(11)\quad \left[\begin{array}{ll}
\text{ANC} & \left[\begin{array}{ll}\text{ACTIVE} & \left[\begin{array}{ll}\text{LIST} & \boxed{1}\\ \text{LAST} & \boxed{4}\end{array}\right]\\ \text{INERT} & \boxed{0}\end{array}\right]\\
\text{ARGS} & \left\langle \left[\text{ANC}\left[\begin{array}{ll}\text{ACTIVE} & \left[\begin{array}{ll}\text{LIST} & \boxed{3}\\ \text{LAST} & \boxed{4}\end{array}\right]\\ \text{INERT} & \left[\begin{array}{ll}\text{LIST} & \boxed{2}\\ \text{LAST} & \boxed{3}\end{array}\right]\end{array}\right]\right], \left[\text{ANC}\left[\begin{array}{ll}\text{ACTIVE} & \left[\begin{array}{ll}\text{LIST} & \boxed{1}\\ \text{LAST} & \boxed{2}\end{array}\right]\\ \text{INERT} & \boxed{0}\end{array}\right]\right]\right\rangle
\end{array}\right]$$

So far, we have only addressed the issue of anchor percolation, but we still need to specify how anchors are inserted into nonlocal features in the first place. In order to avoid spurious ambiguity between in-situ modification and "extraposed" modification of the same head, I will suggest that nonlocal anchors are never introduced below the maximal sign within a head projection. As our main concern here is relative clause extraposition, we will concentrate on nominal anchors only. To achieve this, we will use an additional feature SELF, again a difference list, whose single member is the anchor of the noun phrase (KEY|INDEX and KEY|HANDEL) for saturated nominal signs and the empty difference list for all other signs. We can then adapt our Anchor Percolation Principle as follows:

$$(12)\quad \left[\begin{array}{ll}
\text{ANC} & \left[\begin{array}{ll}\text{ACTIVE} & \left[\begin{array}{ll}\text{LIST} & \boxed{0}\\ \text{LAST} & \boxed{4}\end{array}\right]\\ \text{INERT} & \left[\begin{array}{ll}\text{LIST} & \boxed{5}\\ \text{LAST} & \boxed{7}\end{array}\right]\end{array}\right]\\
\text{ARGS} & \left\langle \left[\text{ANC}\left[\begin{array}{ll}\text{ACTIVE} & \left[\begin{array}{ll}\text{LIST} & \boxed{3}\\ \text{LAST} & \boxed{4}\end{array}\right]\\ \text{INERT} & \left[\begin{array}{ll}\text{LIST} & \boxed{2}\\ \text{LAST} & \boxed{3}\end{array}\right]\\ \text{SELF} & \left[\begin{array}{ll}\text{LIST} & \boxed{0}\\ \text{LAST} & \boxed{1}\end{array}\right]\end{array}\right]\right], \left[\text{ANC}\left[\begin{array}{ll}\text{ACTIVE} & \left[\begin{array}{ll}\text{LIST} & \boxed{1}\\ \text{LAST} & \boxed{2}\end{array}\right]\\ \text{INERT} & \left[\begin{array}{ll}\text{LIST} & \boxed{6}\\ \text{LAST} & \boxed{7}\end{array}\right]\\ \text{SELF} & \left[\begin{array}{ll}\text{LIST} & \boxed{5}\\ \text{LAST} & \boxed{6}\end{array}\right]\end{array}\right]\right]\right\rangle
\end{array}\right]$$

I have so far focussed on binary branching structures only, however, the DFKI German HPSG grammar also makes use of unary rules in the syntax, mostly for traceless introduction of SLASH. Following the rationale adopted for binary phrase structures that only linear intervention of a (right) daughter can make an anchor active for relative clause attachment, it is clear that unary rules cannot activate any inert anchors, for principled reasons. Thus, I will propose that, in these structures, the ANC lists are projected unaltered from daughter to mother, as captured by the following constraint:

$$(13)\quad \left[\begin{array}{ll}
\text{ANC} & \left[\begin{array}{ll}\text{ACTIVE} & \left[\begin{array}{ll}\text{LIST} & \boxed{0}\\ \text{LAST} & \boxed{1}\end{array}\right]\\ \text{INERT} & \left[\begin{array}{ll}\text{LIST} & \boxed{2}\\ \text{LAST} & \boxed{3}\end{array}\right]\end{array}\right]\\
\text{ARGS} & \left\langle \left[\text{ANC}\left[\begin{array}{ll}\text{ACTIVE} & \left[\begin{array}{ll}\text{LIST} & \boxed{0}\\ \text{LAST} & \boxed{1}\end{array}\right]\\ \text{INERT} & \left[\begin{array}{ll}\text{LIST} & \boxed{2}\\ \text{LAST} & \boxed{3}\end{array}\right]\end{array}\right]\right]\right\rangle
\end{array}\right]$$

Once we have set up percolation of indices in this way, we can formulate phrase structure schemata for local and nonlocal relative clause attachment. Whereas the former will be taken care of by standard Head-Adjunct structures, we will postulate a distinct phrase structure schema for nonlocal, extraposed modifiers. Given that retrieval of some meber from a list can only be done in a step-by-step fashion in the underlying formalism (LKB/PET), I will assume that attachment of an extraposed relative clause will introduce an anchor into the TO-BIND|ANC list, which will subsequently be bound to a member in INH|ANC|ACTIVE by means of recursive application of unary retrieval rules.[3]

$$(14)\quad \left[\begin{array}{ll}
\text{ANC} & \left[\text{TO-BIND}\ \ \langle\rangle\right]\\
\text{ARGS} & \left\langle \left[\text{ANC}\left[\begin{array}{ll}\text{TO-BIND} & \left\langle\boxed{1}\right\rangle\\ \text{ACTIVE} & \left[\begin{array}{l}\text{LIST}\left\langle\boxed{1},\ldots\right\rangle\\ \text{LAST}\ \langle\rangle\end{array}\right]\end{array}\right]\right]\right\rangle
\end{array}\right]$$

[3]It is hoped that retrieval rules will at some point become obsolete once the MRS implementation in LKB supports underspecification for intersective modifier attachment.

(15)
$$\begin{bmatrix} \text{ANC} & \begin{bmatrix} \text{TO-BIND} & \left\langle \boxed{1} \right\rangle \\ \text{ACTIVE} & \begin{bmatrix} \text{LIST} & \boxed{2} \\ \text{LAST} & \boxed{3} \end{bmatrix} \end{bmatrix} \\ \text{ARGS} & \left\langle \begin{bmatrix} \text{ANC} & \begin{bmatrix} \text{TO-BIND} & \left\langle \boxed{1} \right\rangle \\ \text{ACTIVE} & \begin{bmatrix} \text{LIST} & \begin{bmatrix} \text{FIRST} & [\,] \\ \text{REST} & \boxed{2} \end{bmatrix} \\ \text{LAST} & \boxed{3} \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

While the first schema performs the actual retrieval, i.e. binding of INDEX and HANDEL values, the second schema above will recurse over the list of active anchors, making available non-initial members for binding. The way the inheritance of anchors has been set up, structurally and superficially more recent anchors will be accessed first. Obligatoriness of retrieval can easily be ensured by requiring fully saturated sentential signs to have an empty ANC|TO-BIND list.

3 Beyond German

Now that that we set up a basic system that permits integration of a Kiss-style approach to relative clause extraposition while avoiding the introduction of spurious ambiguity associated with the original proposal, we may now go on and explore how the current two-step approach can be incorporated into the grammar of languages other than German. Despite the fact that relative clause extraposition is cross-linguistically a quite pervasive phenomenon, none of the large scale LKB/PET grammar implementation efforts currently provide an analysis: apparently, neither the Lingo English Resource Grammar (ERG) (Copestake and Flickinger, 2000), nor the German (Müller and Kasper, 2000; Crysmann, 2003) and Japanese (Siegel, 2000; Siegel and Bender, 2002) grammars developed at DFKI.[4] [5] As a consequence, this phenomenon is not addressed in the HPSG Grammar Matrix (Bender et al., 2002) either, the current version of which has been distilled mainly from the Lingo ERG.

Although extraposition in German is less restricted than, e.g., in English where extraposition from topicalised constituents appears to be ruled out, English nevertheless licenses extraposition from a variety of structural contexts, e.g. subjects, objects, or even fronted wh-constituents.

(16) a. She met a soldier at the party that she really likes.

b. Who do you know that you can really trust? (Kiss, 2003)

Leaving aside for the moment the issue of how the introduction of anchors can or should be restricted here, it is evident that the percolation principles presented above are actually stated independently of any concrete SELF value. Thus, provided the only difference between the scope of extraposition in English and German should lie with the source of anchor percolation, the present approach to anchor *percolation*, could probably be carried over unmodified.

One of the major differences between German and English phrase structure certainly is the degree of canonicity w.r.t. head-dependent order. In German, we can observe quite a good deal of alternation here (think of pre- and postpositions), whereas English tends to be more strongly head-initial, although head-final structures can be found, most notably with Head-Subject structures. Thus, English, just like German, permits new anchors to be introduced on either daughter of a binary schema, left or right. As a consequence, the situations to be controlled for are structurally fully identical.

In our analysis of the source of the spurious ambiguity problem in German, we observed that the problem will always arise whenever local attachment and attachment to a percolated anchor may derive identical surface strings. The solution we have proposed therefore builds on a notion of linear intervention: anchors that are right-peripheral stay inert, anchors which are not, become active.

[4]The statement regarding RC extraposition in the Lingo ERG is the based on a quick inspection of the grammar source code (version form autumn 2002), as well as a handful of tests.

[5]The only implementation done so far in the context of large scale HPSG grammar engineering is Stefan Müller's Babel grammar (Müller, 2002). However, the implementation there builds on the availability of discontinuity, meaning that the analysis cannot be carried over to formalisms which assume continuous constituents only.

An implementation of RC extraposition along the lines described in this paper is currently being developed for the German LKB/PET grammar at DFKI.

The way this is stated is actually pretty much independent of any functional distinctions such as the position of the head or the type of dependent. As a result, we can directly carry over the solution developed for German to any LKB/PET grammar that only distinguishes between binary and unary rule types.[6] Having established the versatility of the current approach in a cross-lingual context, one might wonder though, whether a two-step approach to anchor percolation will always be necessary. As far as I can foresee, the division between inert and active anchors will always make sense, as long as (locally) right-peripheral nodes can introduce anchors. A language where this may not be the case will be canonically head-final, and, furthermore, will either disallow extraposition from extracted dependents, or else, not feature extraction at all. If all these criteria are met in a language, the distinction between active and inert anchors could be conflated, obviating the need for the ANC|INERT list in a grammar of that language. Of course, even under these special circumstances, the approach described here should give the right results, even if unaltered, since a proper specification of ANC|SELF values will guarantee that the ANC|INERT list of right-peripheral daughters will then simply be empty. For the case of SVO languages, such as English, however, where anchors can be introduced initially and finally in a local tree, the utility of the present approach should be evident.

Before closing our discussion, let us briefly return to the difference in restrictiveness that can be observed for relative clause extraposition in German and English.

One such difference has already been addressed above: owing to the fact that extraposed vs. local attachment to *pre*positional phrases or the NPs they contain is empirically indistinguishable, the current approach already blocks the extraposed analysis as a spurious ambiguity. As postpositions are rather uncommon in English, the unavailability of extraposed structures analoguous to (6) is already accounted for by the very mechanisms that rule out spurious ambiguity in (9).

The other major difference relates to extraposition from extracted constituents: in German, extraposition is possible with fronted wh-phrases as well as topicalised constituents, whereas in English, extraposition from topicalised phrases is barred.

(17) * Micro brews, I like that are located around the bay area.

(18) Den Mann hat sie gesehen, den ich
the man has she seen who I
gestern getroffen hatte.
yesterday met had

'She saw the man who I had met yesterday.' (Kiss, 2003)

In general, extracted constituents are not entirely unproblematic for a theory of extraposition: as the latter process is known to be upward bound (Ross, 1967), i.e., typically clause-bound, and percolation of indices may not descend the tree, modification of an antecedent outside the local clause is not trivial. There are, in principle, two ways to resolve this issue: either, anchors are inserted at the gap site, or else, some other mechanism is invoked to relocate binding of the UDC into the clausal domain and percolate the anchor from there (see below).

For (Kiss, 2003), the first strategy is not viable, as local/extraposed attachment is formally non-distinct there, being subsumed under the principle of Generalised Modification. Thus, in traceful formalisms/theories local attachment of a relative clause to a trace cannot be controlled for, deriving the ungrammatical (19).

(19) * [Den Mann]$_i$ hat sie [t_i den ich
the man has she who I
gestern traf] gesehen
yesterday met seen

(Kiss, 2003)

Comparing the constructions featuring extraposition from fronted constituents in English and German, (Kiss, 2003) observed that, in both languages, it is V2 structures that permit such extraposition, whereas English topicalisation, failing to trigger subject-verb inversion, does not appear

[6] The extension to ternary rule types should be trivial, though, as every non-right daughter will actually pattern with the left daughter in a binary rule.

to license extraposition from fronted constituents either. He therefore suggests that the phrase structure of wh-fronting in English is much alike to that of V2 structures in German, which he believes not to be instances of the Head-Filler schema. Instead, he argues, the nonlocal dependency is bound by the inverted verb in second position, which then subcategorises, via SPR, for the wh-constituent in sentence-initial position, thereby assimilating the analysis of wh-fronting in English to that of *tough*-constructions. As an important side-effect, binding of the NONLOCAL|SLASH is moved from the head-filler structure, which is actually outside the local clause onto the head of the clause. This move makes it possible to introduce the anchor of the phrase subcategorised via SPR into the ANCHORS set of the finite verb, and, accordingly, percolate it within the domain of that head, i.e. the local clause. For topicalisation in English, (Kiss, 2003) maintains the classic analysis in terms of head-filler structures, by-passing the head of the clause. Consequently, the filler's anchor will not be introduced in the local clause, and therefore, extraposition from topicalised constituents in English is correctly ruled out.

Although the analysis of extraposition in English proposed by (Kiss, 2003) is actually quite neat, one may wonder though, whether integration of the present refined proposal regarding anchor percolation will necessitate major adaptation of any grammar that features extraposition from extracted constituents? Fortunately, given the properties of the present approach (i.e., the distinction between local and extraposed attachment) and the absence of traces from the underlying formalism (LKB/PET), we are not faced with any strong commitment as to where anchors of constituents must be introduced in order to feature in nonlocal dependencies: while an analysis along the lines of (Kiss, 2003) will always be possible, provided the extraposition facts in a particular language can be related to the same phrase structural properties as in English or German, the approach suggested here, however, is equally compatible with anchor introduction at the gap site, thereby ensuring a high degree of portability. Taking the case of English as an example, integration of relative clause extraposition would thus involve two things: first, unification of the above schemata for anchor percolation with all binary and unary rules that are not involved in anchor retrieval, and second, specification of the conditions on anchor introduction. If a quick and painless integration is all that is called for, it will be sufficient to segregate slash-introduction (lexical) rules according to the value of the slashed element's NONLOC|QUE value: if empty, no anchor for the slashed constituent should be introduced into ANC|SELF (topicalisation), if not, the HANDEL and INDEX features of the slashed element will get inserted as an anchor by the slash introduction rule. Thus, even under such an admittedly simple scenario, the number of slash introduction rules for a language such as English would only increase from 3 to 6. Needless to say, that for a language such as German, where all UDCs pattern alike w.r.t. extraposition, the single added cost lies with the attachment and retrieval rules, i.e. 2 in total.

4 Conclusion

Although the present approach is admittedly slightly less general than the proposal after which it is modelled (Kiss, 2002) — we need two distinct phrase structure schemata instead of one — it appears to us that this is the price to pay, if we want to avoid proliferation of spurious ambiguity in an implemented grammar of German. On the other hand, the current approach to controlling spurious ambiguity is general enough to be applicable to almost any other language that features relative clause extraposition, as it is formulated independently of the position of the head, which is variable in German. Moreover, we have found that besides head-projection paths it is left-branching structures which are a notorious source for spurious ambiguity in extraposition contexts. We therefore expect the current results to be of some use for an implementation of RC extraposition in more strictly head-initial languages as well. We have shown that the current proposal can be carried over unmodified to account for similar phenomena in English, without incurring any substantial changes to existing implemented grammars. Taken together, the independence of head position and the ease of integration with existing grammars will hopefully make this approach qualify as a candidate for integration into language independent core grammars, such as the HPSG Matrix Grammar initiative (Bender et al., 2002).

References

Emily M. Bender, Dan Flickinger, and Stephan Oepen. 2002. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammar. In John Carroll, Nelleke Oostdijk, and Richard Sutcliffe, editors, *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, Taipei.

Ulrich Callmeier. 2000. PET — a platform for experimentation with efficient HPSG processing techniques. *Journal of Natural Language Engineering*, 6(1):99–108.

Ann Copestake and Dan Flickinger. 2000. An open-source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the Second conference on Language Resources and Evaluation (LREC-2000)*, Athens.

Ann Copestake. 2001. *Implementing Typed Feature Structure Grammars*. CSLI Publications, Stanford.

Berthold Crysmann. 2003. On the efficient implementation of German verb placement in HPSG. ms., DFKI, Saarbrücken.

Andreas Kathol. 1995. *Linearization-Based German Syntax*. Ph.D. thesis, Ohio State University.

Andreas Kathol. 2000. *Linear Syntax*. Oxford University Press, Oxford.

Frank Keller. 1995. Towards an account of extraposition in HPSG. In *Proceedings of the Ninth Meeting of the European ACL*, Dublin. Association for Computational Linguistics.

Tibor Kiss. 2002. Semantic constraints on relatice clause extraposition. *Natural Language and Linguistic Theory*, forthcoming.

Tibor Kiss. 2003. Phrasal typology and the interaction of topicalization, wh-movement, and extraposition. In Jong-Bok Kim and Stephen Wechsler, editors, *Proceedings of the 9th International Conference on Head-driven Phrase Structure Grammar, Kyung Hee University, Seoul, 5–7 August, 2002*, Stanford. CSLI Publications.

Stefan Müller and Walter Kasper. 2000. HPSG analysis of German. In Wolfgang Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*, pages 238–253. Springer, Berlin.

Stefan Müller. 1996. The babel-system: An HPSG Prolog implementation. In *Proceedings of the 4th International Conference on the Practical Application of Prolog*, pages 263–277, London.

Stefan Müller. 1999. *Deutsche Syntax — deklarativ*. Linguistische Arbeiten. Niemeyer, Tübingen.

Stefan Müller. 2001. Continuous or discontinuous constituents. Technical report, DFKI GmbH, Saarbrücken.

Stefan Müller. 2002. Continuous or discontinuous constituents? A comparison between syntactic analyses for constituent order and their processing systems. to appear in *Research on Language and Computation*.

John Nerbonne. 1994. Partial verb phrases and spurious ambiguities. In John Nerbonne, Klaus Netter, and Carl Pollard, editors, *German in Head-Driven Phrase Structure Grammar*, number 46 in Lecture Notes, pages 109–150. CSLI Publications, Stanford University.

John R. Ross. 1967. *Constraints on Variables in Syntax*. Ph.D. thesis, MIT.

Melanie Siegel and Emily M. Bender. 2002. Efficient deep processing of Japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization. Coling 2002 Post-Conference Workshop*, Taipei.

Melanie Siegel. 2000. HPSG analysis of Japanese. In Wolfgang Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer, Berlin.

Compositional semantics in a multilingual grammar resource

Dan Flickinger and Emily M. Bender
Center for the Study of Language and Information
Stanford University, Stanford, CA 94305 USA
danf@csli.stanford.edu, bender@csli.stanford.edu

1 Introduction

The Matrix grammar starter-kit (Bender et al., 2002) is a language-independent core grammar designed to facilitate the rapid initial development of grammars for natural languages, with foundations solid enough to support steady expansion to broad coverage of the linguistic phenomena in these languages. Such grammars are particularly valuable because they can assign semantic representations to linguistic input, providing the foundation for applications which require natural language understanding. As such, a central component of the Matrix is the collection of resources it contains for simplifying the implementation of semantic composition within each language and supporting the development of a standardized description language for meaning representations, which can provide an effective interface for practical applications. The resources in the Matrix further enable the meaning representations to keep pace as the syntactic analyses of a grammar grow in complexity. In this paper we present the methodology and mechanisms employed for semantic composition in the Matrix grammar starter-kit, using Minimal Recursion Semantics (MRS) with grammars written in the Head-driven Phrase Structure Grammar (HPSG) framework.

2 Background

The goal of the Matrix grammar starter-kit is to provide the necessary definitions of core linguistic types at a level of generality which enables quick specialization to encode the additional basic grammatical constraints for a particular language. With this language-specific tuning, it should be possible to construct a grammar within an afternoon which can be used to parse nontrivial sentences of a given language, then use that same implementation as the basis for the development over time of a semantically precise, broad-coverage grammar. The existing Matrix release also includes software links and parameter settings for one particular grammar development system, the LKB (Copestake, 2002), which includes an efficient parser and generator, but grammars built on the Matrix can be read and used by a number of other parsers (cf. Oepen et al. (2003)).

The Matrix is constructed within the formal system of typed feature structures defined in Carpenter (1992), using the single operation of unification to build phrases from the words and phrases they contain. Minimal Recursion Semantics (MRS: Copestake et al. (1999)) was designed to enable semantic composition using only this same unification of typed feature structures, producing for each phrase or sentence a description of the meaning representation sufficient to support logical inference. The type definitions for signs in the Matrix include a semantic component which is an implementation of MRS, and more specifically of the elaboration of a semantic algebra for MRS presented in Copestake et al. (2001). Very briefly, this framework assigns a flat semantic representation to each word or phrase, consisting of

1. RELS - a bag of atomic predications introduced by lexical entries or by syntactic constructions, each with a “handle” (used to express scope relations) and one or more roles;

2. HCONS - a set of handle constraints which reflect syntactic limitations on possible scope relations among the atomic predications;

3. HOOK - a group of distinguished externally visible attributes of the atomic predications in RELS, used in combining the semantics of this sign with the semantics of other signs.

3 Implementation

3.1 Architecture of Matrix CONT

The overall architecture of a sign in the Matrix follows closely the definition in Pollard and Sag (1994), with a SYNSEM attribute consisting of (at least) CAT and CONT attributes, for syntactic and semantic constraints, respectively. The value of CONT is a feature structure of type *mrs*, with three attributes HOOK, RELS, and HCONS, encoding the three parts of an MRS representation outlined above. The values of RELS and HCONS are implemented for convenience as lists, since they represent information that is accumulated as one moves up the tree (see below), while the value of HOOK is a feature structure that introduces attributes for each of the four externally visible semantic elements that a sign may present for further composition. These four attributes are as follows:

1. LTOP - the handle of the relation in RELS with highest scope.

2. INDEX - instance or event variable introduced by the lexical semantic head, analogous to lambda variable.

3. E-INDEX - an additional event variable required for some signs like gerunds or predicative phrases, which need to expose both a nominal instance and an event, for control and modification.

4. XARG - the semantic index of the sign’s externally visible argument, if any (typically the subject of a verb phrase or other controlled complement).

All of the above is implemented in the type *mrs*, defined below, where *individual* is the supertype to both *event* and *index*:

(1)
$$\textit{mrs:}\left[\begin{array}{ll}\text{HOOK} & \left[\begin{array}{ll}\multicolumn{2}{l}{\textit{hook}}\\ \text{LTOP} & \textit{handle}\\ \text{INDEX} & \textit{individual}\\ \text{E-INDEX} & \textit{individual}\\ \text{XARG} & \textit{individual}\end{array}\right]\\ \text{RELS} & \text{diff-list}\\ \text{HCONS} & \text{diff-list}\end{array}\right]$$

3.2 Semantics principles for phrases

With every word or phrase providing a semantics which consists of these three parts (HOOK, RELS, and HCONS), the principles of semantic composition in phrase structure rules can be stated (and implemented) quite elegantly, following the definitions in Copestake et al. (2001):

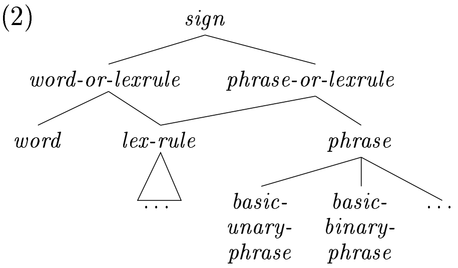
1. The value for RELS on the mother of a phrase is the result of appending the RELS values of all of its daughters.

2. The value for HCONS on the mother of a phrase is the result of appending the HCONS values of all of its daughters.

3. The value for HOOK on the mother of phrase is identified with the HOOK value of its semantic head daughter, where each phrase type uniquely determines which of the daughters is the semantic head.

In the Matrix (version 0.4, March 14, 2003), principles 1 and 2 are implemented as constraints on a few high-level types (*lex-rule*, *basic-unary-phrase* and *basic-binary-phrase*) within the *sign* subhierarchy (sketched in (2)), such that they are inherited by all phrases and lexical rules. In addition, the type *phrase-or-lexrule* identifies the mother's HOOK with the HOOK of the semantics provided by the rule itself (the value of a feature called C-CONT; see §3.4 below). More specialized phrase types identify the HOOK of the C-CONT with the HOOK of the head or non-head daughter.

(2)



Principles 1 and 2 require the accumulation of RELS and HCONS values from daughter to mother in a phrase. The values of these features are implemented as difference lists (typed feature structures which bear values for two attributes LIST and LAST), allowing us to state the accumulation of values using the same single operation of unification of typed feature structures. (3) shows the constraints on the type *basic-unary-phrase*, including the 'diff-list appends' that implement principles 1 and 2.

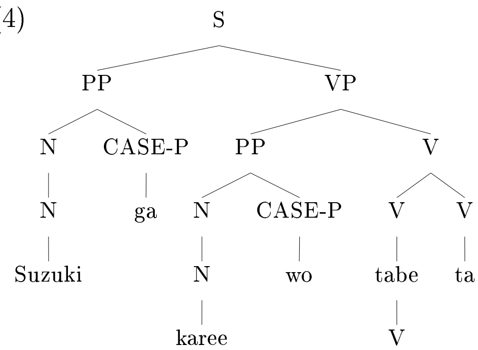
(3)

$$\begin{bmatrix} \textit{basic-unary-phrase} & \\ \text{SYNSEM.LOCAL.CONT} & \begin{bmatrix} \text{RELS} & \begin{bmatrix} \text{LIST} & \boxed{1} \\ \text{LAST} & \boxed{2} \end{bmatrix} \\ \text{HCONS} & \begin{bmatrix} \text{LIST} & \boxed{4} \\ \text{LAST} & \boxed{5} \end{bmatrix} \end{bmatrix} \\ \text{C-CONT} & \begin{bmatrix} \text{RELS} & \begin{bmatrix} \text{LIST} & \boxed{3} \\ \text{LAST} & \boxed{2} \end{bmatrix} \\ \text{HCONS} & \begin{bmatrix} \text{LIST} & \boxed{6} \\ \text{LAST} & \boxed{5} \end{bmatrix} \end{bmatrix} \\ \text{ARGS} & \left\langle \begin{bmatrix} \ldots & \text{CONT} \begin{bmatrix} \text{RELS} & \begin{bmatrix} \text{LIST} & \boxed{1} \\ \text{LAST} & \boxed{3} \end{bmatrix} \\ \text{HCONS} & \begin{bmatrix} \text{LIST} & \boxed{4} \\ \text{LAST} & \boxed{6} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

3.3 Composing heads and arguments

All remaining particulars about the semantics of a phrase are determined by the reentrancies for semantic attributes specified by that phrase, or by the signs which unify in as daughters of the phrase. For example, in the Japanese sentence in (4),[1] the verb *tabe-* ('eat') identifies the HOOK|INDEX of its subject with the ARG1 of the 'taberu' relation, and the HOOK|INDEX of its object with the ARG2.

(4)



Suzuki-NOM curry-ACC eat-PAST

The rules which build the S and the VP (Japanese-specific specializations of the head-subject and head-complement rules) identify the *synsem* of the PP with the relevant valence requirement (e.g., COMPS) of the verbal head daughter. Since the *synsem* includes the CONT value, a cascade of identities of the familiar kind relates the indices of *Suzuki* and *karee* with the correct roles in the 'taberu' relation. That is, the verb's lexical entry imposes both syntactic and semantic constraints on the synsems of its subject and complement, and the syntactic rules propagate those constraints to the phrases corresponding to the subject and complement, gathering up the semantic relations introduced by the lexical entries and establishing via simple unification the reentrancies that link the semantics of these phrases as intended. Once the verbal

[1]The tree given is a simplification of the tree assigned to this sentence by the JACY grammar (Siegel, 2000; Siegel and Bender, 2002).

head has combined with its complement PP, the semantic variable introduced by that PP's noun as its (externally visible) index is now unified with the appropriate semantic argument position in the relation introduced by the verb. The subject-head rule has an analogous effect on the semantic index of the subject PP, ensuring that it is identified with the other argument position in the verb's relation. The resulting MRS is shown in (5):

(5)

$$\left[\begin{array}{l}
\textbf{mrs} \\
\text{HOOK} \begin{bmatrix} \text{LTOP } \textbf{h1} \\ \text{INDEX } \textbf{e2} \end{bmatrix} \\
\text{RELS} < \begin{bmatrix} \textbf{prpstn} \\ \text{LBL } \textbf{h1} \\ \text{MARG } \textbf{h4} \end{bmatrix}, \begin{bmatrix} \textbf{named} \\ \text{LBL } \textbf{h7} \\ \text{ARG0 } \textbf{x8} \\ \text{NAMED } \textbf{“Suzuki”} \end{bmatrix}, \\
\quad \begin{bmatrix} \textbf{def} \\ \text{LBL } \textbf{h10} \\ \text{ARG0 } \textbf{x8} \\ \text{RSTR } \textbf{h11} \\ \text{BODY } \textbf{h12} \end{bmatrix}, \begin{bmatrix} \textbf{karee} \\ \text{LBL } \textbf{h14} \\ \text{ARG0 } \textbf{x16} \end{bmatrix}, \\
\quad \begin{bmatrix} \textbf{udef} \\ \text{LBL } \textbf{h17} \\ \text{ARG0 } \textbf{x16} \\ \text{RSTR } \textbf{h18} \\ \text{BODY } \textbf{h19} \end{bmatrix}, \begin{bmatrix} \textbf{taberu} \\ \text{LBL } \textbf{h21} \\ \text{ARG0 } \textbf{e2} \\ \text{ARG1 } \textbf{x8} \\ \text{ARG2 } \textbf{x16} \end{bmatrix} > \\
\text{HCONS} < \begin{bmatrix} \textbf{qeq} \\ \text{HARG } \textbf{h4} \\ \text{LARG } \textbf{h21} \end{bmatrix}, \begin{bmatrix} \textbf{qeq} \\ \text{HARG } \textbf{h11} \\ \text{LARG } \textbf{h7} \end{bmatrix}, \\
\quad \begin{bmatrix} \textbf{qeq} \\ \text{HARG } \textbf{h18} \\ \text{LARG } \textbf{h14} \end{bmatrix} >
\end{array}\right]$$

In this representation, the *prpstn* relation (indicating the illocutionary force of the utterance, see Ginzburg and Sag (2000)) bears the top handle of the sentence as its LBL value, and takes as its argument the label of the *taberu* relation, mediated by a *qeq* constraint[2] in the HCONS list of handle constraints to allow for possible intervening quantifiers. Two quantifiers are introduced in this example: a definite quantifier for the proper name and an underspecified one for the bare noun. Each of the two noun relations introduces a variable as its ARG0 value, with each of these variables bound by the appropriate quantifier relation, which also identifies its restrictor value as the label of that noun's relation (again mediated by *qeq* constraints). Since quantifier scope is left underspecified in the grammar, the BODY attributes of the two quantifiers are left with unbound values.[3] The ARG0 variable introduced by *Suzuki* is identified with the ARG1 of the *taberu* relation, and that of *karee* with the ARG2 role. These radically underspecified role names in the *taberu* relation will be interpreted by the grammar-specific semantic interface to indicate that Suzuki does the eating and the curry gets eaten.

3.4 Semantic contributions of constructions

Since some phrase types may introduce semantic content which is not drawn from any of the daughters of the phrase, the MRS framework provides an attribute for phrasal signs called C-CONT (for construction content), which behaves with respect to the semantics principles just as though it were another daughter of the phrase (see, for example, (3) above). C-CONT is implemented in the Matrix as a top-level attribute of phrases and lexical rules, introduced on the type *phrase-or-lexrule*. Like CONT, its value is of type *mrs*.

If a phrase does not introduce any additional semantic content of its own, the values for the attributes RELS and HCONS in C-CONT will be empty lists, so unary and

[2] *Qeq* constraints state that the HARG and LARG are of equal scope, unless a quantifier scopes in between, in which case the HARG outscopes the quantifier which outscopes the LARG.

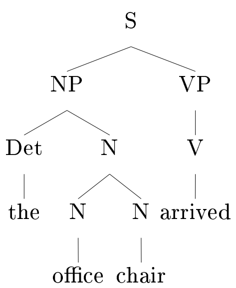
[3] The underspecified MRS in (5) is compatible with two fully resolved MRSs, representing the two possible scopings of the two quantifiers.

binary phrases can safely always append these values to those supplied by the syntactic daughters. Likewise, the HOOK value of a phrase is always identified with its C-CONT's value for HOOK, where for most phrases this HOOK in C-CONT will simply be identified with that of one of the daughters of the phrase, namely the semantic head daughter.

One example of a phrase type in the Matrix that does introduce its own semantic content is the type for (non-relative) clauses (*non-rel-clause*), which introduces a relation encoding the illocutionary force of the clause. Such relations (e.g., *prpstn-rel*, mentioned above) are of type *message*, following the analysis in Ginzburg and Sag (2000).

We illustrate construction-introduced semantic content further with the treatment of noun-noun compounds in the English Resource Grammar (Flickinger, 2000), which uses the same principles of semantic construction as the Matrix toolkit. In the analysis of the sentence *the office chair arrived*, the phrase *office chair* is built using a syntactic rule specifically for noun-noun compounds, and this rule introduces a generic two-place relation *n-n-cmpnd* which relates the variables introduced by the two nouns. The syntactic structure is sketched in (6), where the head-specifier rule is used to combine the determiner and the compound noun, while the head-subject rule combines the full NP with the verb phrase *arrived*. The corresponding MRS semantics is shown in (7):

(6) *The office chair arrived*



(7)

$$\begin{bmatrix} \textbf{mrs} \\ \text{HOOK} \begin{bmatrix} \text{LTOP } \textbf{h1} \\ \text{INDEX } \textbf{e2} \end{bmatrix} \\ \text{RELS} < \begin{bmatrix} \textbf{prpstn} \\ \text{LBL } \textbf{h1} \\ \text{MARG } \textbf{h4} \end{bmatrix}, \begin{bmatrix} \textbf{def} \\ \text{LBL } \textbf{h10} \\ \text{ARG0 } \textbf{x8} \\ \text{RSTR } \textbf{h11} \\ \text{BODY } \textbf{h12} \end{bmatrix}, \\ \begin{bmatrix} \textbf{chair} \\ \text{LBL } \textbf{h7} \\ \text{ARG0 } \textbf{x8} \end{bmatrix}, \begin{bmatrix} \textbf{office} \\ \text{LBL } \textbf{h14} \\ \text{ARG0 } \textbf{x16} \end{bmatrix}, \begin{bmatrix} \textbf{udef} \\ \text{LBL } \textbf{h17} \\ \text{ARG0 } \textbf{x16} \\ \text{RSTR } \textbf{h18} \\ \text{BODY } \textbf{h19} \end{bmatrix}, \\ \begin{bmatrix} \textbf{n-n-cmpnd} \\ \text{LBL } \textbf{h7} \\ \text{ARG1 } \textbf{x16} \\ \text{ARG2 } \textbf{x8} \end{bmatrix}, \begin{bmatrix} \textbf{arrive} \\ \text{LBL } \textbf{h21} \\ \text{ARG0 } \textbf{e2} \\ \text{ARG1 } \textbf{x8} \end{bmatrix} > \\ \text{HCONS} < \begin{bmatrix} \textbf{qeq} \\ \text{HARG } \textbf{h4} \\ \text{LARG } \textbf{h21} \end{bmatrix}, \\ \begin{bmatrix} \textbf{qeq} \\ \text{HARG } \textbf{h11} \\ \text{LARG } \textbf{h7} \end{bmatrix}, \begin{bmatrix} \textbf{qeq} \\ \text{HARG } \textbf{h18} \\ \text{LARG } \textbf{h14} \end{bmatrix} > \end{bmatrix}$$

Note that this MRS representation for the English example is in many respects similar to that of the earlier Japanese example, again introducing two noun relations and supplying a quantifier relation to bind each of these two variables. Here, however, the two noun variables are identified with the ARG1 and ARG2 attributes of the *n-n-cmpnd* relation, and the variable for the head noun *chair* is also the value of the single argument of the *arrive* relation. The *n-n-cmpnd* relation is introduced by the grammar in the RELS attribute of the C-CONT of the grammar rule for noun-noun compounds, which also identifies the assignments of the two nominal instance variables (supplied by its two daughters) to the ARG0 and ARG1 attributes of that *n-n-cmpnd* relation. The relevant constraint on the grammar rule is sketched in (8):

(8)
$$\begin{bmatrix} \text{HEAD-DTR...HOOK} & \begin{bmatrix}\text{INDEX } \boxed{1}\end{bmatrix} \\ \text{NON-HEAD-DTR...HOOK} & \begin{bmatrix}\text{INDEX } \boxed{2}\end{bmatrix} \\ \text{C-CONT} & \begin{bmatrix}\text{RELS.LIST} & \left\langle \begin{bmatrix} \textit{n-n-cmpnd} \\ \text{ARG1 } \boxed{1} \\ \text{ARG2 } \boxed{2}\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}$$

As discussed above, general principles of semantic composition that are encoded in the Matrix types ensure that rule-specific relations are gathered up along with the relations supplied by the daughters of the rule, and that the appropriate external semantic hooks (the LTOP and INDEX values) are identified on the phrase itself, ready for further composition.

3.5 Lexical rules

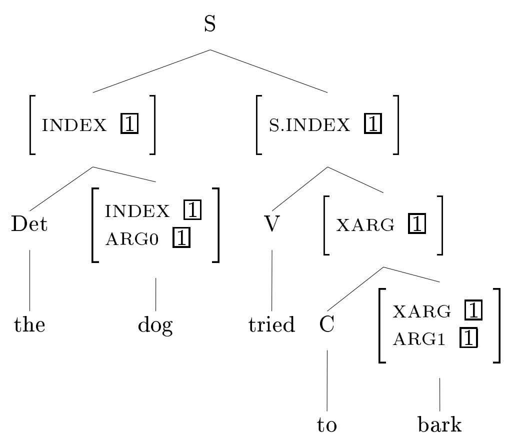
Lexical rules are treated in the Matrix as a particular type of unary rule, in most respects like syntactic unary rules, though lexical rules are prevented from interleaving with syntactic rules. Thus semantic composition for lexical rules is implemented using the same principles as outlined above for syntactic phrases. A lexical rule may or may not introduce semantic content of its own; if it does, that content is found in the C-CONT attribute of the rule and is combined with the semantic content of the (single) daughter (the 'input' to the lexical rule) by those same principles.

Note that this approach imposes a strong constraint on the directionality of lexical rules in the Matrix, since the principles of composition guarantee monotonic accumulation of atomic predications, so no semantic content from a daughter in a phrase or lexical rule is ever lost. For example, a lexical rule relating the causative/inchoative alternation in English for verbs like *break* will have to treat the inchoative lexical entry as the 'input' to the lexical rule (that is, the daughter), and the semantically richer causative lexical entry as the 'output' (the SYNSEM value of the lexical rule).

3.6 External arguments

In the examples above, we made reference to two of the *hook* attributes, LTOP and INDEX, both of which play a crucial role in the semantic construction of every phrase. A third attribute, XARG, is relevant for control phenomena such as equi and raising, since it identifies the semantic index of a phrase's external argument (usually the subject of a verb phrase). Identifying this property of a phrase as part of the hook allows our general principles of semantic composition to make this attribute visible for control of subject-unsaturated complements (VPs, predicative PPs, etc.) and also for agreement even at the sentence level, as for example in tag questions in English (Bender and Flickinger, 1999). An example using this XARG attribute in composition is given in (9), with the lexical type for subject-equi verbs given in (10).

(9) The dog tried to bark



(10) Type for subject-equi verbs like *try*

$$\begin{bmatrix} \textit{subj-equi-verb} & \\ \text{SUBJ} & < \begin{bmatrix}\text{HOOK.INDEX } \boxed{1}\end{bmatrix} > \\ \text{COMPS} & < \begin{bmatrix}\text{HOOK.XARG } \boxed{1}\end{bmatrix} > \\ \text{CONT.RELS} & < \begin{bmatrix}\text{ARG1 } \boxed{1}\end{bmatrix} > \end{bmatrix}$$

Here the lexical entry for the verb *try* identifies its VP complement's semantic external argument (XARG value) with its subject's semantic index (INDEX value), and further identifies that index with the appropriate role (the ARG1) in the lexical relation introduced by the verb. The MRS semantics constructed for this example is given in (11).

(11)

$$
\left[\begin{array}{l}
\textbf{mrs} \\
\text{HOOK} \begin{bmatrix} \text{LTOP } \textbf{h1} \\ \text{INDEX } \textbf{e2} \end{bmatrix} \\
\text{RELS} < \begin{bmatrix} \textbf{prpstn} \\ \text{LBL } \textbf{h1} \\ \text{MARG } \textbf{h4} \end{bmatrix}, \begin{bmatrix} \textbf{def} \\ \text{LBL } \textbf{h10} \\ \text{ARG0 } \textbf{x8} \\ \text{RSTR } \textbf{h11} \\ \text{BODY } \textbf{h12} \end{bmatrix}, \\
\quad \begin{bmatrix} \textbf{dog} \\ \text{LBL } \textbf{h7} \\ \text{ARG0 } \textbf{x8} \end{bmatrix}, \begin{bmatrix} \textbf{try} \\ \text{LBL } \textbf{h21} \\ \text{ARG0 } \textbf{e2} \\ \text{ARG1 } \textbf{x8} \\ \text{ARG2 } \textbf{h22} \end{bmatrix}, \\
\quad \begin{bmatrix} \textbf{prpstn} \\ \text{LBL } \textbf{h22} \\ \text{MARG } \textbf{h23} \end{bmatrix}, \begin{bmatrix} \textbf{bark} \\ \text{LBL } \textbf{h24} \\ \text{ARG0 } \textbf{e3} \\ \text{ARG1 } \textbf{x8} \end{bmatrix} > \\
\text{HCONS} < \begin{bmatrix} \textbf{qeq} \\ \text{HARG } \textbf{h4} \\ \text{LARG } \textbf{h21} \end{bmatrix}, \\
\quad \begin{bmatrix} \textbf{qeq} \\ \text{HARG } \textbf{h11} \\ \text{LARG } \textbf{h7} \end{bmatrix}, \begin{bmatrix} \textbf{qeq} \\ \text{HARG } \textbf{h23} \\ \text{LARG } \textbf{h24} \end{bmatrix} >
\end{array}\right]
$$

The construction of this representation is the result of the same general principles of semantic composition presented above. The head-complement rule unifies the verb *try*'s constraints on its complement with those of the VP phrase *to bark*, which results in the identification of the XARG value of that VP with the INDEX of the subject of *try*. The constraints on *try*'s subject are propagated up to the verb phrase *tried to bark* from the head-daughter *tried* by the head-complement rule, and the semantics of this verb phrase preserve the semantic properties of its daughters, including the desired re-entrancies with the subject index. Hence when the head-subject rule combines *the dog* with *tried to bark*, the syntactic and semantic constraints of the noun phrase are unified with those in the SUBJ attribute of the verb phrase, resulting in the identification of the ARG0 value introduced by it dog with the ARG1 values in both the *try* relation and the *bark* relation.

3.7 Future work: lexical semantics

The primary focus of the current Matrix machinery for the syntax-semantics interface is on the definitions of the semantic principles for composition in the construction of syntactic phrases. In its current early form, the Matrix provides only minimal support for defining the semantics of lexical entries, though work is underway to add a partial hierarchy of lexical types to express generalizations about some standard subcategorization phenomena, including the linking of syntactic arguments to semantic roles in atomic predications. This will build on the set of primary types of lexical relations (*arg1-rel*, *arg2-rel*, *arg12-rel*, ...) presently provided. These types introduce a perhaps surprisingly sparse set of role names that are nonetheless meant to be (nearly) exhaustive.

The task of linking syntactic arguments to these semantic roles is still left as an exercise for the developers of each individual grammar. Developers are encouraged to avoid the introduction of any additional role names, and to avoid the use of multiple relations for individual lexical entries for open-class lexeme types, so that grammars being developed using the current Matrix will be compatible with the next release aimed at carrying more of the burden of lexical type definitions.

4 Conclusion

In this paper, we have illustrated how the implementation of the syntax-semantics interface in the Matrix provides support for the rapid development of broad-coverage precision grammars. We have identified the principles of semantic composition adopted in the Matrix using the framework of Minimal Recursion Semantics, and illustrated their interaction with the semantic properties introduced by lexical entries and syntactic constructions. Initial experiments applying the Matrix to grammars of Norwegian, Italian and Greek, some of which are reported on in this workshop, have validated the benefits of the machinery already provided for semantic composition in speeding up the development of semantically rich grammars. The experience of these grammar writers also underscores the significant potential benefit of planned extensions to the Matrix including a lexical type hierarchy, which will help to sustain the standardization across implementations and languages which is one of the chief goals of the Matrix.

5 Acknowledgements

We have benefited greatly in this work from conversations with Ann Copestake, Alex Lascarides, Stephan Oepen, and Ivan Sag on general issues of semantic composition; from insights provided by Melanie Siegel and Berthold Crysmann who are developing broad-coverage grammars for Japanese and German, respectively; and from our interactions with the resourceful and cooperative developers of Matrix-based grammars for Norwegian and Italian.

References

Emily Bender and Dan Flickinger. 1999. Peripheral constructions and core phenomena. In Gert Webelhuth, Andreas Kathol, and Jean-Pierre Koenig, editors, *Lexical and Constructional Aspects of Linguistic Explanation*. CSLI Publications, Stanford.

Emily Bender, Dan Flickinger, and Stephan Oepen. 2002. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, Taipei, Taiwan.

Bob Carpenter. 1992. *The Logic of Typed Feature Structures*. Cambridge University Press, Cambridge, UK.

Ann Copestake, Daniel P. Flickinger, Ivan A. Sag, and Carl Pollard. 1999. Minimal Recursion Semantics. An introduction.

Ann Copestake, Alex Lascarides, and Dan Flickinger. 2001. An algebra for semantic construction in constraint-based grammars. In *Proceedings of the 39th Meeting of the Association for Computational Linguistics*, Toulouse, France.

Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Publications, Stanford, CA.

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6 (1) (Special Issue on Efficient Processing with HPSG):15 – 28.

Jonathan Ginzburg and Ivan A. Sag. 2000. *Interrogative Investigations: The form, meaning and use of English interrogatives*. CSLI Publications, Stanford, CA.

Stephan Oepen, Daniel Flickinger, J. Tsujii, and Hans Uszkoreit, editors. 2003. *Collaborative Language Engineering. A Case Study in Efficient Grammar-based Processing*. CSLI Publications, Stanford, CA.

Carl Pollard and Ivan A. Sag. 1994. *Head-driven Phrase Structure Grammar*. Chicago University Press, Chicago.

Melanie Siegel and Emily M. Bender. 2002. Efficient deep processing of japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and Standardization at the 19th Internation Conference on Computational Linguistics*, Taipei, Taiwan.

Melanie Siegel. 2000. HPSG analysis of Japanese. In Wolfgang Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer, Berlin.

NorSource - an exercise in the *Matrix* grammar-building design

Lars Hellan
Department of Studies in Language and Communication, NTNU
lars.hellan@hf.ntnu.no

Petter Haugereid
Department of Studies in Language and Communication, NTNU
petterha@hf.ntnu.no

Abstract

As a step in operationalizing a design of maximal compatibility between grammars of distinct languages (as far as theoretical frameworks allow), the *HPSG Grammar Matrix* offers a shared 'top' typed feature structure, for being worked into existing grammars based on HPSG, and for being used as a starting point for the construction of new grammars. Currently, this 'top' structure is essentially an excerpt from the *ERG* (*English Resource Grammar*), offering insights encoded in this large coverage grammar for immediate incorporation in other grammars. It comprises 'technical' types (such as types for list definitions, for construction of MRS, and others), and grammar-structural types of some generality. Extending the latter aspect to more specific types, and supplementing it with lexical types, while maintaining the system as a common denominator for an interestingly large set of grammars, is one of the main tasks presently pursued within this enterprise. This paper presents some constructs that are currently part of the NorSource computational grammar of Norwegian, and which may be candidates for incorporation in an 'extended' Matrix.

1 Introduction

The *NorSource* Grammar ('Norwegian HPSG Resource Grammar') is a computational grammar for Norwegian currently under construction at NTNU, Trondheim.[1] It is based on the linguistic framework *HPSG (Head-Driven Phrase Structure Grammar)* (Pollard and Sag 1994, Sag and Wasow 1999), and uses the Grammar Engineering platform *LKB* (*Linguistic Knowledge Builder*) (Copestake 2002).[2] [3] From its inception (Spring 2002), it has used the *HPSG Grammar Matrix* (Bender, Flickinger and Oepen 2003)[4], henceforth *Matrix*, based essentially on the *ERG* (*English Resource Grammar*) (Flickinger 2000). As of Fall 2002, it is part of the Multilingual Grammar Engineering sub-consortium of the EU project *Deep Thought*,[5] devoted to the development and accommodation of the Matrix relative to the participating grammar projects.

The initial versions of the Matrix provide types inducing the basic sign feature geometry, types for the essential kinds of lexical rules (lexeme-to-word, lexeme-to-lexeme, with and without in

1 Actively involved participants include: Dorothee Beermann, Kaja Borthen, Petter Haugereid, Lars Hellan, Stephan Oepen, Siri Simonsen, and Jostein Ven. The project's URL is: http://www.ling.hf.ntnu.no/forskning/norsource/ Financial support comes in part from the EU project *Deep Thought* (see footnote 5).

2 http://www-csli.stanford.edu/~aac/lkb.html

3 The project has a parallel in - and is by 2-3 years anteceded by - the *LFG*-based project *NorGram* at the University of Bergen; cf. http://www.hf.uib.no/i/LiLi/SLF/Dyvik/norgram/ and http://www.ling.uib.no/~victoria/norgram/

4 Version 0.4; version 0.1 was released in January 2002.

5 (http://www.eurice.de/deepthought/index.html), funded by the IST programme, with participantion from Saarland University, Cambridge University, University of Sussex, NTNU, Stanford University, CELI (Torino) and XtraMind (Saarbrücken).

duction of affixational morphology) and syntactic rules (binary vs. unary, headed vs. non-headed, head final vs. head initial; head-complement, head-subject, head modifier,...), types inducing MRS representations, types for list manipulation, and more, all hypothesized as necessary (or at least very useful) in any core HPSG grammar. In particular, a chosen format of MRS representations is supposed to constitute part of all Matrix-based grammars. The present scope of the Matrix still leaves large domains of specification open, such as lexical specification, argument structure information, degrees of strictness of constituent order, and much more. Desiderata for the development of the system therefore include (i) designing possible 'universals' for lexical specification; (ii) defining parameters of linguistic variation, through devices such as boolean choices with regard to given feature values, and availability vs. absence of certain subtypes of generally needed types. Such choices may easily lead to systems that are not compatible with all conceivable grammars, but still accommodate large enough subsets to serve the purpose. (In principle, one may of course come to a point where simply converting one concrete grammar into another (as possibly, when going from one Mainland Scandinavian language to another) might be easier than designing a common abstract 'matrix'; discussion of actual instances of such a choice situation is as yet premature.) In the present note, we describe some constructs that are currently part of, or potentially part of, the NorSource design, and which may be candidates for incorporation in an 'extended' Matrix.

2 Valence specification

At a somewhat lower level of generality, we have encoded a phenomenon of *fields of constituent order* as manifest in the (Mainland) Scandinavian languages, through defining three lists as relevant for the distribution of complements of verbs, namely (in this order), ACOMPS, NCOMPS, and ICOMPS. A(nnex)COMPS contains light pronouns as clustered together with the verb; N(uclear)COMPS contains those complements whose mutual order is largely fixed, and in between which adverbial adjuncts can generally not intervene; I(nterspersable)COMPS contains complements whose relative order is less strict, and in between which adverbial adjuncts *may* be interspersed. ICOMPS consists largely of PPs (with either nominal or sentential governees), NCOMPS consists of objects, predicatives, and 'particles'. Objects (direct and indirect) may appear either on NCOMPS or ACOMPS, according to their 'weight'; lexical rules are supposed to interrelate the options (see section 5).

As 'domains', the objects represented by these features are quite different from, e.g., the German 'scrambling domains',[6] since they represent fixed orders (only items in ICOMPS may internally have some freedom of sequential order). They perhaps have their closest counterparts in 'clitic domains' as found, e.g., in Romance and Balkan languages, even though only the items in ACOMPS bear any resemblance to what are commonly recognized as 'clitics'.[7] The distinction NCOMPS vs. ICOMPS indeed might be applied also to a grammar of English (even though the ERG so far doesn't have it) and the other Mainland Scandinavian languages, but not to other Germanic languages like German, nor to Romance languages. Hence while the phenomena seem clearly identifiable, and the representing mechanisms generalizable beyond the grammar of Norwegian and possibly candidates for inclusion in a 'matrix' system at some level, this level is clearly not one of higher generality.

The lists mentioned above will, like COMPS and SUBJ in standard HPSG grammars, technically serve for encoding (degree of) satisfaction of 'valence'. Still, they are a bit impractical for the purpose of identifying bearers of specific grammatical functions (like 'direct object'), since a given object can occur either on the ACOMPS or on the NCOMPS list, and a subject can be on either the SUBJ list or (in inverted constructions) ACOMPS or NCOMPS. To the extent that grammatical functions (GFs) need to be identified for reference, this suggests that at least some GFs be represented 'directly'. For instance, it is arguable, on the basis of the indefiniteness requirement in presentational constructions (cf. (Hellan 2003), that the GF 'direct object' in Norwegian be represented as an individual GF.

[6] cf., e.g., Reape 1993, Kathol 2000, and others.
[7] For their characterization, proposals of 'lite' status as in Abeillé and Godard 1997 are quite relevant.

In the argument structure of verbs in Norwegian, it would be linguistically plausible to distinguish (at least) five GFs: 'subject', 'direct object', 'indirect object', 'oblique', and 'predicative'. For grammar engineering concerns, it is at the present point not obvious where the balance should lie between richness of 'navigation points', favoring many GFs, and parsimony with regard to types and attributes. We presently employ GF attributes corresponding to all five GFs: SUBJECT, D(irect)OBJECT, I(ndirect)OBJECT, OBL and PREDIC(ative).[8] An inventory like this may seem easily generalizable across languages, although in considering such a GF system for inclusion in the Matrix system, factors like its overall processing cost will first have to be assessed.

Having now recognized two aspects of the notion 'valence' in NorSource, these are encoded under different attributes, VAL(ence) for the field-cum-saturation aspect, and Q(ualitative)VAL for the GF aspect, both in turn grouped together under the feature CAT. A simple, partial illustration is given in figure 1.

```
trans-arg1-2-le
STEM < "beundre" >
SS  trans-arg1-2-verb
    LOC  CAT  HEAD verb
              VAL  SUBJ <#0 [np-synsem
                             LOC.CONT.HOOK.INDEX #1]>
                   NCOMPS<#3 [np-synsem
                              LOC.CONT.HOOK.INDEX #4]>
                   ICOMPS ⟨ ⟩
                   ACOMPS ⟨ ⟩
              QVAL [SUBJECT #0
                    DOBJECT #3]
         CONT [HOOK.INDEX #2
               RELS <! #5 !>]
         KEYS.KEY #5 [arg1-2-rel
                      PRED "beundre-v-rel"
                      ARG0 #2
                      ARG1 #1
                      ARG2 #4]
INFLECTION v1
```

Figure 1. AVM (simplified) for the verb lexeme *beundre* ('admire').

3 Semantics interface

Any constituent, at any rank, has a semantic specification introduced by the attribute CONT(ent)., whose value belongs to the type *mrs* (for 'minimal recursion semantics', cf. (Copestake et al. 2001), (Copestake et al. 1999)). Among the attributes introduced by this type is RELS, which is a 'bag' (formally construed as a *difference list*) of all those 'elementary predications' (EPs) which are contained, lexically or constructionally, in the constituent in question. An EP is specified by a *handle*, by which the position of the predication in a hierarchy of relative *scopes* can be indicated, and *arguments* as induced by the type of *relation* expressed by the predicate; for quantifiers, there is also a 'bound variable'. Most EPs are introduced in the entry of a lexial item, and most commonly, a lexical entry introduces just one EP (as a value of KEY). The value of KEY is copied over to RELS (as indicated by the reentrancy in fig. 1). Correspondences between semantic and syntactic structures are stated through the use of a type hierarchy subsumed by a type *synsem*, introduced by the attribute SYNSEM, defining which *cat* types can match which *relation* types. For example, there will be one *synsem* type - *trans-arg1-2-verb* - declaring the combination of the *relation* type *arg1-2-rel* and the *cat* type *trans-cat* as licit, and specifying which syntactic arguments link up to which semantic arguments; a licensed combination is illustrated in fig.1.[9]

For verbs, adjectives, prepositions, adverbs and nouns, features introduced by *relation* generally include ARG0, as representing an event index or other index reflecting the 'validity span' of the predicate in question, and ARG1, as the 'external argument' of the predicate. ARG2 represents any other argument in addition to ARG1, and for verbs, a possible further argument (expressed by indirect objects in ditransitive constructions) is encoded as ARG3. Heuristically, these arguments of a verb may be interpreted to the effect of the attribute ARG1 introducing

[8] These are not unlike attributes used in LFG - cf. (Bresnan 2000, Dalrymple 2002).

[9] The apparent redundancy residing in *trans-arg1-2* occurring sufffixed by *-le* as a *sign* subtype (in a hierarchy we are not showing here) and by *-verb* as a *synsem* subtype, modelled after ERG, reflects a mechanism for encoding optionality (cf. (Flickinger 2000a)).

items with an *initiating* role relative to the event/situation expressed by the verb, the attribute ARG2 introducing items with a *patient/theme* role, and the attribute ARG3 introducing items with a *receiving, benefactive* or *experiencing* role. It should be noticed, though, that such 'proto-roles' are not *definitionally* linked to the attributes in question: an alternative interpretation of ARG1, ARG2 and ARG3 is for instance that of semantically unspecified roles correlated with 'basic' GFs associated with the verb stem in question, i.e., reflecting its argument frame prior to valence-affecting lexical rules (see below). On this option, there is still a choice between, for a verb like *break* as in *the window broke*, to count its sole argument as an ARG2, reflecting a traditional approach to 'unaccusativity', or as an ARG1. The issue partly depends on whether one wants a 'monotonic' representation of alternation pairs like *the window broke* vs. *John broke the window*. In a grammar with only minimal encoding of individual GFs, using ARG1 to encode 'subject' may be natural,[10] while with a design like the one for NorSource described above, it is possible to put the ARG features more in the service of semantic distinctions. This is the option taken in NorSource.

On either of the positions sketched, there is a clear isomorphy between items appearing in a syntactic frame specification and items in a *relation* specification. Thus, what is introduced under KEY/RELS gives little room for, e.g., semantic decomposition: what gets expounded are connections only between referents of *overt* arguments, however possibly such that also *non-overtly* expressed roles played by these arguments are represented (like in 'equi-', 'raising-', and other 'argument sharing' constructions).

Apart from the representations of argument sharing and of possible 'role' status of arguments, a third locus for display of semantic information of grammatical relevance is under an attribute *SORT*, which is found on the feature path `SS.LOCAL.HOOK.CONT.INDEX.SORT`. For instance, a 'directional' secondary predication construction like (1a) has the alternative order (1b) (accommodated by a lexical rule of 'Particle Preposing' - cf. section 5). While *ut* in (1a) may have either a directional or a resultative reading, in (1b) it can only have a resultative reading:

(1) a. Jon kastet ballen ut
Jon threw ball-the out
b. Jon kastet ut ballen

In the RELS/KEY specification of *kaste* ('throw'), the frame option accommodating an item like *ut* in (1) will have an attribute *ARGX* in addition to ARG1 and ARG2, and the elementary predication associated with *ut* is linked to ARGX. ARGX in principle accommodates a range of constituent types, with GF either OBL or PREDIC, and a diversity of semantic values. Since the lexical rule creating the frame accommodating (1b) is sensitive to whether the content of ARGX is *result*, the path `SS.LOCAL.HOOK.CONT.INDEX.SORT` in this specification is given a continuation `...SORT.ARGX-VALUE result`. With such a reference to resultativity, the Particle Preposing rule can be restricted to apply only when *ut* has a resultative reading (cf. section 5).

As a representation of resultativity as such, the above is of course rather impoverished. For expression of richer semantic information, the Matrix will provide a separate component called *SEM-I*; cf. Copestake and Flickinger 2003 for a preliminary overview. As a possible alternative or supplement, under development in the NorSource environment, the syntactically very simple Norwegian grammar *NorMRS* (Beermann 2003),[11] with a full MRS structure, explores the use of a feature *SEM-TYPE*, where richer lexical semantic analysis is encoded in such a way as to supplement the specifications derived from KEY on the MRS RELS list. (Possible anticipated use of such information is, for instance, as *interlingua* in machine translation applications.)

4 Reuse of lexical resources

From the computational lexicography projects TROLL (1987-1992) and NorKompLeks ('NKL'; 1996-2001), both at NTNU, resources are avail

10 Notice that although there is some flexibility in the interpretation of 'ARG1', 'ARG2', etc.,, this flexibility will not extend to the point of allowing 'ARG1' to be understood as 'the *linearly first* argument', 'ARG2' as 'the linearly second argument', and so forth: for instance, an ARG3 is linked to an NCOMPS position precedeing that of an ARG2.

11 Based largely on Copestake 2001, and serving as an 'experimental semantic' grammar.

able for reuse both in the area of verb classification and in the assignment of inflection.

4.1. Porting verb classifications

NKL provides a classification of about 10,000 verb lemmas, encoded in a theory neutral classification system developed in TROLL, whereby a verb occurrence type (a 'frame') is specified in terms of each argument's syntactic category, its GF and its thematic role. In TROLL, a system of lexical rules would interrelate multiple frames available for a single verb, while in NKL, all frame possibilities are encoded in a 'flat' design, with about 100 possible frame types assigned a 'frame code', such as 'intrans1', 'intrans2' and so forth; one given verb may thus be listed with many such coded frames. The classification offered by NKL being by far the largest in terms of the number of verb lemmas classified, a table has been defined by which NKL codes are correlated with verb lexeme types as definable in the NorSource system described above; for instance, the NKL type `trans1` will correspond to the NorSource type *trans-arg1-2-le* illustrated in fig. 1. Applying an algorithm changing the NKL codes into the NorSource codes for all the verbs, the NorSource lexicon can in turn thereby accommodate the NKL classification.

Underlying many of the distinctions made in NKL are semantically flavored factors like unaccusativity vs. intransitivity, agentivity vs. non-agentivity, and experiencer status vs. non-experiencer status. In the NorSource format described above, they will be reflected either in choices between 'ARG1', 'ARG2' and 'ARG3', or in specifications under the SORT atribute.

A wider applicability of this transfer algorithm will depend on whether computational lexical resources of a similar format obtain for other languages. The algorithm has no effects for the design of the NorSource grammar itself, except to some extent providing single verbs with a 'flat' set of entries rather than using lexical rules, to most effectively define the porting algorithm.

4.2 Porting inflectional specification

Another type of information transferred from NKL to NorSource is that of *inflectional paradigms* for word classes such as verb, noun and adjective. In NKL, each word is associated with a paradigm class label, which represents specific spell-change operations for the various inflection categories, one set of changes for each paradigm. For instance, the verb *smile* (smile) has the inflectional code *v2*, which indicates that the verb has the form *smile* in infinitive, in imperative the last letter will be deleted (*smil*), in present tense an *r* will be added (*smiler*), in preteritum the last letter will be deleted and *te* will be added (*smilte*), and so forth. Altogether, Norwegian verbs have 12 inflectional forms, essentially with a 'one-morph' design. The following system operationalizes such paradigm class labels in inflectional specification.

First, we introduce the feature INFLECTION on the type *lexeme*, thereby making sure that all lexeme entries have this feature. The value of the feature will in every single lexeme be a type that corresponds to the inflectional code assigned to the lexeme in NKL For instance, the lexical entry for *smile* 'smile' looks like this:

$$\begin{bmatrix} \text{intrans-arg1-le} \\ \text{INFLECTION v2} \\ \text{STEM < "smile" >} \\ \text{SYNSEM.LOCAL.CONT.RELS <! } [\text{PRED "smile-rel"}] \text{ !>} \end{bmatrix}$$

A type hierarchy of inflectional codes is introduced, part of it illustrated below. Under *v-infl* we have types that correspond to the inflections represented in NKL (*inf-0*, *imp-1-0*, *pres-r*, *pret-1-te* …). We let each inflectional code type (i.e., INFLECTION value, or 'paradigm code') inherit from the relevant inflectional types. In the type hierarchy below we see that the paradigm type *v2* inherits from *inf-0*, *imp-1-0*, *pres-r*, *pret-1-te* and *ppart-1-t*.

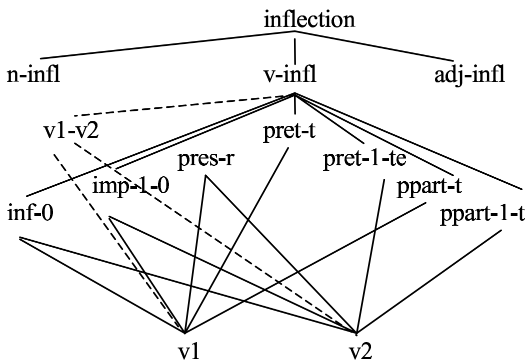


Figure 2: Type system interconnecting 'paradigm types' with types representing inflectional operations.

Operationally, inflection is done through a unary lexical rule with a lexeme as its daughter and a word as its mother, specified by the addition of a suffix. In the inflectional rule rendered below, the value of DTR is the daughter (what stands after the '+' sign represents the added suffix):

$$\begin{bmatrix} \text{verb-pres\_infl\_rule} \\ \text{STEM} \langle \#1 + \text{r} \rangle \\ \text{SYNSEM } \#2 \\ \text{DTR} \begin{bmatrix} \text{verb-lxm} \\ \text{STEM} \langle \#1 \rangle \\ \text{SYNSEM } \#2 \\ \text{INFLECTION pres-r} \end{bmatrix} \end{bmatrix}$$

This rule requires of the daughter (the lexeme) that the value of the feature INFLECTION be compatible with the type *pres-r*. In the type hierarchy in fig. 2, we see that the inflectional pattern type *v2* is compatible with *pres-r* (*v2* is a subtype of *pres-r*), therefore the lexeme "smile" (which has the feature INFLECTION *v2*) can undergo this rule.

Some lexemes in Norwegian may follow more than one inflectional pattern; for instance, the verb *bane* (make one's way) may follow both inflectional patterns *v1* and *v2*:

bane – ban – baner – banet – banet (v1)
bane – ban – baner – bante – bant (v2)

In these cases we make a type *v1-v2* which inherits from *v-infl*, and which has *v1* and *v2* as its subtypes. Since types which have common subtypes are compatible, the system lets a lexeme with the feature INLECTION v1-v2 undergo all rules that require an inflectional code which is a supertype of *v1* or *v2*.

5 Lexical rules.

Phenomena such as passive, light pronouns, subject-verb inversion, and particle preposing are encoded by lexical rules, and briefly described in the following. After each example of an instantiating sentence follows a statement of the relevant rule. In the (simplified) rule presentations, the *input* of the rule is given as 'ARGS' (another attribute introducing daughters).

- *Passivization*: In Norwegian, a passive sentence can have either the direct object ((3)) or the indirect object ((2)) of the corresponding active sentence as subject; also an expletive (*det*) can function as subject ((4)).

(2) Gutten ble gitt en is.
boy-the was given an ice-cream
'The boy was given an ice-cream.'

$$\begin{bmatrix} \text{passive-ncomps1-lrule} \\ \text{RULE passive-constr} \\ \text{SYNSEM.LOCAL.CAT.VAL} \begin{bmatrix} \text{SUBJ} \langle \#1 \rangle \\ \text{NCOMPS } \#2 \end{bmatrix} \\ \text{ARGS} \left\langle \begin{bmatrix} \text{RULE base} \\ \text{SYNSEM.LOCAL.CAT.VAL} \begin{bmatrix} \text{SUBJ} \langle \text{np-synsem} \rangle \\ \text{NCOMPS} \langle \#1 \,.\, \#2 \rangle \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

(3) En is ble gitt gutten.
an ice-cream was given boy-the
'The boy was given an ice-cream.'

$$\begin{bmatrix} \text{passive-ncomps2-lrule} \\ \text{RULE passive-constr} \\ \text{SYNSEM.LOCAL.CAT.VAL} \begin{bmatrix} \text{SUBJ} \langle \#2 \rangle \\ \text{NCOMPS} \langle \#1 \,.\, \#3 \rangle \end{bmatrix} \\ \text{ARGS} \left\langle \begin{bmatrix} \text{RULE base} \\ \text{SYNSEM.LOCAL.CAT.VAL} \begin{bmatrix} \text{SUBJ} \langle \text{np-synsem} \rangle \\ \text{NCOMPS} \langle \#1 \,.\, \#2 \,.\, \#3 \rangle \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

(4) Det ble gitt gutten en is
it was given the boy an ice-cream
'The boy was given an ice-cream.'

$$\begin{bmatrix} \text{passive-presentational-lrule} \\ \text{RULE passive-constr} \\ \text{SYNSEM.LOCAL.CAT.VAL} \begin{bmatrix} \text{SUBJ} \langle \text{expletive-synsem} \rangle \\ \text{NCOMPS } \#1 \end{bmatrix} \\ \text{ARGS} \left\langle \begin{bmatrix} \text{RULE base} \\ \text{SYNSEM.LOCAL.CAT.VAL} \begin{bmatrix} \text{SUBJ} \langle \text{np-synsem} \rangle \\ \text{NCOMPS } \#1 \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

- *Particle preposing*: Some particles can occur both before and after the object, as illustrated in (1) above.

$$\begin{bmatrix} \text{particle-preposing-lrule} \\ \text{RULE particle} \\ \text{SYNSEM.LOCAL} \begin{bmatrix} \text{CAT.VAL.NCOMPS} \langle \#1, \#2 \rangle \\ \text{CONT.HOOK.INDEX.SORT.ARGX-VALUE result} \end{bmatrix} \\ \text{ARGS} \left\langle \begin{bmatrix} \text{RULE particle-dtr} \\ \text{SYNSEM.LOCAL.CAT.VAL.NCOMPS} \langle \#2, \#1 \rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

- *Inversion*: In yes-no-questions (5) and topicalization the subject comes after the verb.

(5) Liker Kari isen?
likes Kari ice-cream-the
'Does Kari like the ice-cream?'

$$\left[\begin{array}{l} \text{inversion-lrule} \\ \text{RULE inversion} \\ \text{SYNSEM.LOCAL.CAT.VAL} \left[\begin{array}{l}\text{SUBJ} <\ > \\ \text{NCOMPS} < \#1 \,.\, \#2 > \end{array}\right] \\ \text{ARGS} < \left[\begin{array}{l} \text{RULE inversion-dtr} \\ \text{SYNSEM.LOCAL.CAT.VAL} \left[\begin{array}{l} \text{SUBJ} < \#1 \text{ np-synsem} > \\ \text{NCOMPS} < \#2 > \end{array}\right]\end{array}\right] > \end{array}\right]$$

- *Light pronouns*: Norwegian light pronouns have a slightly different syntactic distribution than other (heavy) NPs (reflected in the adoption of an ACOMPS list). When they are arguments of the finite verb, light pronouns necessarily occur in the position preceding a sentence adverbial like *ikke* ('not') ((6), (7) and (8)), a position normally blocked for heavy NPs.

(6) Kari liker den ikke.
Kari likes it(light) not
'Kari doesn't like it.'

$$\left[\begin{array}{l} \text{acomps1-lrule} \\ \text{RULE acomps} \\ \text{SYNSEM.LOCAL.CAT.VAL} \left[\begin{array}{l}\text{ACOMPS} < \#1 > \\ \text{NCOMPS } \#2 \end{array}\right] \\ \text{ARGS} < \left[\begin{array}{l} \text{RULE acomps-dtr} \\ \text{SYNSEM.LOCAL.CAT.VAL} \left[\begin{array}{l} \text{ACOMPS} <> \\ \text{NCOMPS} < \#1 \,.\, \#2 > \end{array}\right]\end{array}\right] > \end{array}\right]$$

(7) Kari gir ham den ikke.
Kari gives him(light) it(light) not
Kari doesn't give it to him.

$$\left[\begin{array}{l} \text{acomps2-lrule} \\ \text{RULE acomps} \\ \text{SYNSEM.LOCAL.CAT.VAL} \left[\begin{array}{l}\text{ACOMPS} < \#1, \#2 > \\ \text{NCOMPS } \#3 \end{array}\right] \\ \text{ARGS} < \left[\begin{array}{l} \text{RULE acomps-dtr} \\ \text{SYNSEM.LOCAL.CAT.VAL} \left[\begin{array}{l} \text{ACOMPS} <> \\ \text{NCOMPS} < \#1 \,.\, \#2 \,.\, \#3 > \end{array}\right]\end{array}\right] > \end{array}\right]$$

(8) Gir hun ham den ikke?
gives she him it not
'Doesn't she give it to him?'

$$\left[\begin{array}{l} \text{acomps3-lrule} \\ \text{RULE acomps} \\ \text{SYNSEM.LOCAL.CAT.VAL} \left[\begin{array}{l}\text{ACOMPS} < \#1, \#2, \#3 > \\ \text{NCOMPS } \#4 \end{array}\right] \\ \text{ARGS} < \left[\begin{array}{l} \text{RULE acomps-dtr} \\ \text{SYNSEM.LOCAL.CAT.VAL} \left[\begin{array}{l} \text{ACOMPS} <> \\ \text{NCOMPS} < \#1 \,.\, \#2 \,.\, \#3 \,.\, \#4 > \end{array}\right]\end{array}\right] > \end{array}\right]$$

The lexical rules presented here have to apply in a fixed order, with the Passive lexical rules first, then the Particle-preposing rule, then the Inversion rule, and then the Acomps rules. To account for this ordering of rules, we have added a feature RULE on the type *word-or-lexrule*, with a value of a type *construction*. All lexical rules have this RULE constraint. The RULE feature of the daughter of each rule is constrained to be compatible with the RULE feature of only those signs that the rule may take as its input. So the RULE value of the Particle-preposing-lrule's daughter is constrained to be *particle-dtr*, a type which is a subtype of *base* and *passive-constr*.[12] This means that the Particle preposing lexical rule can take base lexemes and lexemes which have undergone a passive lexical rule, as input. In a sentence like (9) we get the longest possible chain of lexical rules:

(9) Kastes det ikke ut en ball?
Throw-PASS it not out a ball
'Isn't there being thrown out a ball?'

inflected word
|
acomps1 lexical rule
|
inversion lexical rule
|
particle preposing lexical rule
|
passive-presentational lexical rule
|
base-lexeme

The lexical rules now presented account for many argument frame possibilities for Norwegian verbs, but not without cost. A verb like *gi* ('give') has two lexical entries, one as a ditransitive verb and one as a transitive verb with an ad

[12] The type hierarchy does in essence what a finite state algorithm could also do, with arcs corresponding to the supertype-subtype relations.

ditional PP complement. When we parse the sentence *Jon ble gitt en is* ('Jon was given an ice-cream') with the current NorSource grammar, the parser will try both entries, in combination with a number of lexical rules. All in all, 37 different valence versions of 'give' will be tried out (only two will give a parse).

As an alternative to this approach, we develop an 'experimental' grammar with radically less lexical rules (NorSource-g2). This grammar lets lexical entries be more underspecified with regard to the order in which arguments are realized. Instead, the phrasal rules impose constraints on the composed sign during the derivation. In constructing such a grammar, we make use of the same ad-hoc feature RULE which we used for lexical rules, but its function now is to constrain which kind of signs a phrase strucure rule allows as its head daughter. Parsing and generation time seem to get significantly reduced with this approach; the significance of this will have to be assessed in an overall evaluation of these options.

6 Concluding remarks

Apart from providing a picture of NorSource in general, and - admittedly to a lesser extent - of the current version of the Matrix, we have (section 2) presented the 'field' constructs A/N/ICOMPS, and the QVAL supplement to the VAL feature, as possible contributions to the Matrix architecture - perhaps not at a most general level, but for an interestingly large set of possible grammars. The choices made by NorSource with regard to 'semantic load' of the ARG attributes (section 3) are also possible within a more general architecture.[13] Illustrations, references to strategic 'side grammars' and reuse of existing resources, have accompanied these proposals (sections 4 and 5). On the building-strategic side, having mainly discussed possible amendments and extensions to the Matrix, let us finally stress the effects of the Matrix even at its present stage for contributing to a quick start and enhanced development of the NorSource grammar, effects which are undoubtedly replicated in other up-starting grammars, and for contributing to cross grammar compatibility, not least in the MRS component.

[13] See also (Borthen 2003) for proposals of a general nature concerning the semantic specifications of nominals, motivated partly from Norwegian, and amenable to inclusion in NorSource as well as, possibly, a Matrix system.

References

Anne Abeillé and Danièle Godard. 1997. French Word Order and Lexical Weight. Ms.

Dorothee Beermann. 2003. NorMRS. NTNU.

Emily M. Bender, Dan Flickinger and Stephan Oepen. 2003. The HPSG Grammar Matrix, version 0.4.

Kaja Borthen. 2003. A grammar component for semantic classes of nominals in HPSG. Workshop on Multilingual Grammar Engineering, ESSLLI 2003.

Joan Bresnan. 2000. *Lexical Functional Syntax.* Blackwell Publishers, Oxford.

Ann Copestake. 2001. Compositional MRS Grammar. Cambridge University.

Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Publications, Stanford.

Ann Copestake and Dan Flickinger. 2003. The semantic interface to the ERG and other LinGO grammars (initial partial draft). Ms, Cambridge University.

Ann Copestake, Dan Flickinger, Ivan Sag and Carl Pollard. 1999. Minimal Recursion Semantics: An introduction' (draft).

Ann Copestake, Alex. Lascarides and Dan Flickinger. 2001. An Algebra for Semantic Construction in Constraint-based Grammars. *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics* (ACL 2001), Toulouse.

Mary Dalrymple. 2001. *Lexical Functional Grammar.* Academic Press.

Dan Flickinger. 2000. The English Resource Grammar. CSLI.

Dan Flickinger. 2000a. On building a more efficient grammar by exploiting types. *Natural Language Engineering* 6(1): 15-28, Cambridge University Press.

Lars Hellan. 2003. The NorSource Grammar - an introduction. Ms, NTNU.

Andreas Kathol. 2000. *Linear Syntax.* Oxford University Press.

Carl Pollard and Ivan Sag. 1994. *Head Driven Phrase Structure Grammar.* Chicago University Press, Chicago.

Mike Reape. 1993. A Formal Theory of Word Order. A Case Study in West Germanic. PhD. diss., Edinburgh.

Ivan Sag and Tom Wasow. 1999. *Syntactic Theory. A Formal Introduction.* CSLI Publications, Stanford.

Multilingual Grammar Development via Grammar Porting

Roger Kim
Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, CA 94304 USA
rkim@parc.com

Mary Dalrymple
Dept. of Computer Science
King's College London
Strand, London WC2R 2LS UK
mary@dcs.kcl.ac.uk

Ronald M. Kaplan
Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, CA 94304 USA
kaplan@parc.com

Tracy Holloway King
Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, CA 94304 USA
thking@parc.com

Hiroshi Masuichi
Corporate Research Center
Fuji Xerox Co.
Kanagawa 259-0157 JAPAN
hiroshi.masuichi@fujixerox.co.jp

Tomoko Ohkuma
Corporate Research Center
Fuji Xerox Co.
Kanagawa 259-0157 JAPAN
ohkuma.tomoko@fujixerox.co.jp

Abstract

In this paper, we investigate using an existing deep Lexical-Functional (LFG) grammar to develop a new one for a typologically similar language. In particular, we ported the Japanese ParGram grammar to Korean with promising results after only two months.

1 Introduction

The Parallel Grammar project (ParGram) is an international collaboration aimed at producing broad-coverage computational grammars for a variety of languages (Butt et al., 1999; Butt et al., 2002). The grammars (currently of English, French, German, Japanese, Norwegian, and Urdu) are written in the framework of Lexical Functional Grammar (LFG) (Kaplan and Bresnan, 1982; Dalrymple, 2001), and they are constructed using a common engineering and high-speed processing platform for LFG grammars, the XLE system (Maxwell and Kaplan, 1993). These grammars, as do all LFG grammars, assign two levels of syntactic representation to the sentences of a language: a surface phrase structure tree (called a *constituent structure* or c-structure) and an underlying matrix of features and values (the *functional structure* or f-structure). The c-structure records the order of words in a sentence and their hierarchical grouping into phrases. The f-structure encodes the grammatical functions, syntactic features, and predicate-argument relations conveyed by the sentence. F-structures are meant to encode a language universal level of analysis, allowing for cross-linguistic parallelism at this level of abstraction.

The ParGram project attempts to test the LFG formalism for its universality and coverage and to see how far parallelism can be maintained across languages; previous ParGram work (and much theoretical analysis) has largely confirmed the universality claims of LFG theory. The f-structures produced by the grammars for similar constructions in each language have the same major functions and features, with minor variations across languages (e.g., the f-structures for French nouns have a gender feature but that distinction is not marked in English f-structures). This uniformity has the computational advantage that the grammars can be used in similar applications and that machine translation (Frank, 1999) can be simplified.

We have found that it takes roughly two person-years of effort to construct for a new language a grammar that approximates existing grammars in terms of coverage and accuracy (see (Riezler et al., 2002) for a discussion of the coverage and accuracy of the current English grammar). This suggests that the deep-grammar construction task is not as difficult as many people have suggested, and indeed may require less effort than would be needed to produce training materials for automatic learning procedures for shallower grammars. Nonetheless, we are exploring methods for reducing the linguistic effort that grammar construction requires. The approach described here investigates the difficulty of converting a grammar of one language into a grammar of a typologically similar language. In this investigation, we started with the ParGram grammar of Japanese and used that as the basis for a grammar of Korean.

Typologically similar (but not necessary genetically related) languages are those that not only allow for similar f-structures (as LFG theory suggests is the case with all languages) but also have similar c-structure to f-structure mappings. Whether or not Japanese and Korean are genetically related (an issue that is in some dispute; see (Sohn, 1999) for some discussion), they are typologically similar in at least the following ways: they both are verb final, have relatively free word order, use postpositions to mark grammatical functions, and exhibit rampant pro drop.[1]

The creation of the current Japanese grammar (Masuichi and Ohkuma, 2003) involved two person-years of work at Fuji Xerox. The grammar has broad coverage, providing parses for 97% of sentences in a real-world test suite with good accuracy. Experiments done to test accuracy of the parses show that the ParGram Japanese grammar is comparable to standard Japanese dependency parsers; however, the ParGram grammar provides linguistically more detailed information than basic dependency relations. The Japanese grammar has 50 annotated phrase structure rules which compile out to finite-state machines with a total of 346 states, 1224 arcs, and 1702 disjuncts.

2 Porting Syntactic Rules

2.1 Direct Porting

The ParGram LFG grammars consist of phrase structure rules that are annotated with information about the corresponding f-structures. The creation of these annotated phrase structure rules forms the bulk of the effort in creating deep grammars for a new language. Thus, if we can port the annotated phrase structure rules for one language into the grammar of another, significant time can be saved. To do this, the two languages must be typologically similar, as is the case of Japanese and Korean.

The word order possibilities required no modification between the Japanese and Korean grammars. The basic verb final order of Japanese could be carried over to Korean, along with free ordering of preceding arguments and adjuncts, including markings for prefered word orders (e.g., subject preceding object). Sample orders covered by the grammars are shown in (1) and (2).

(1) a. Ayuko ga gakusei ni hon wo ageta.
Ayuko NOM student DAT book ACC gave
'Ayuko gave the student a book.' (Japanese)

b. gakusei ni hon wo Ayuko ga ageta.

c. hon wo Ayuko ga gakusei ni ageta.

(2) a. Myungwoni ga haksaeng ehgeh
Myungwoni NOM student DAT

chaek ul juuttda.
book ACC gave
'Myungwoni gave the student a book.' (Korean)

b. haksaeng ehgeh chaek ul Myungwoni ga juuttda.

c. chaek ul Myungwoni ga haksaeng ehgeh juuttda.

The structures for the Korean sentence in (2a) are shown in Figures 1 and 2. The corresponding Japanese structures for (1a) are shown in Figures 3 and 4. The structures are identical except for the lexical items (see section 3 on morphological differences). In the version of the Korean grammar shown here, the system works on Latin transliterated Korean, but it is currently being adapted to use Hangul script.

Similarly, the rules for topicalization could also be ported without modification. In Japanese, noun phrases are marked as topicalized by the postposition *ha*. Topicalized noun phrases may have certain postpositions before the final *ha*, as in (3a). However, nominals with postposition case markers such as *wo*, *ga*, or *no* cannot be topicalized by *ha*. Instead, the postposition is dropped and only

[1]See (Paik and Shirai, 2001) who exploit this similarity in machine translation.

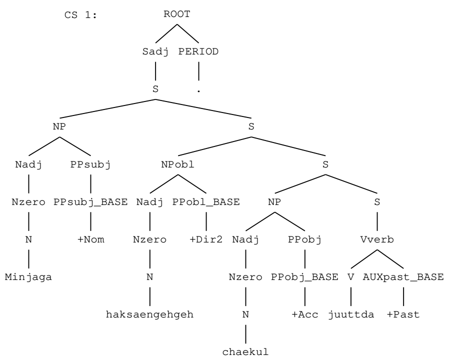


Figure 1: Korean c-structure for (2a)

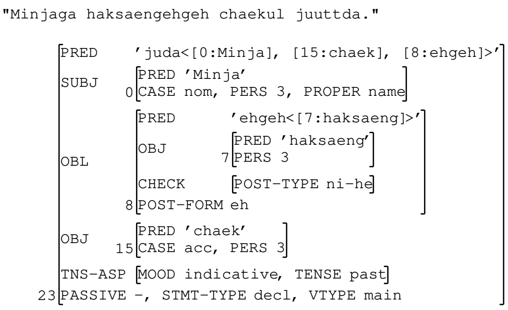


Figure 2: Korean f-structure for (2a)

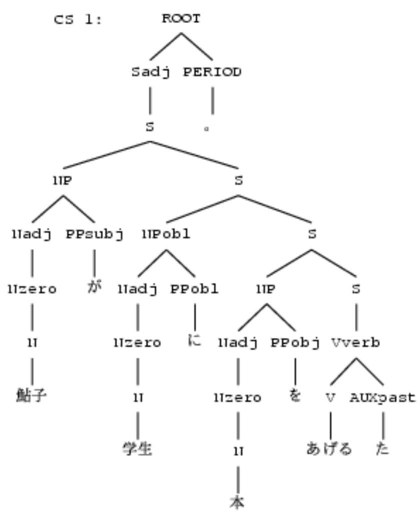


Figure 3: Japanese c-structure for (1a)

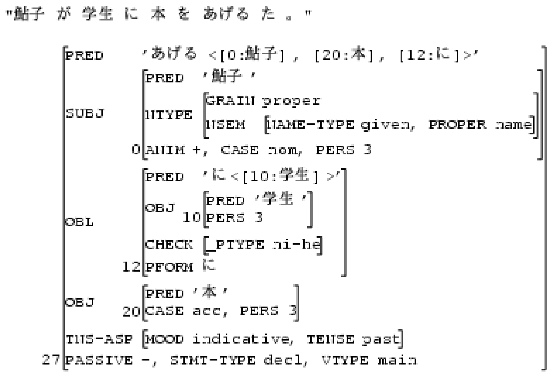


Figure 4: Japanese f-structure for (1a)

ha appears, as in (3c). In addition, these phrases are marked in the f-structure as to their topic status; this f-structure information controls their syntactic distribution in the sentence. The corresponding topicalizing postposition in Korean is *un/nun*, with the allomorph *un* following vowel-final nominals and *nun* following consonant-final nominals, as in (3b). Just as with the Japanese *ha*, the Korean topicalizer also cannot cooccur with postpositions marking the basic grammatical functions, as in (3d).

(3) a. kinoo made ha
yesterday until TOPIC
'until yesterday (topic)' (Japanese)

b. uhjeh kaji nun
yesterday until TOPIC
'until yesterday (topic)' (Korean)

c. *kinoo ga ha $\Longrightarrow$ kinoo ha

d. *uhjeh ga nun $\Longrightarrow$ uhjeh nun

Nominal internal structure was also ported directly from the Japanese grammar to the Korean. This includes the analysis of adjectival, nominal, and postpositional modifiers of the head noun. For example, the rules used to produce the analysis for the Japanese complex nominal in (4a) were ported directly to produce the analysis of the Korean nominal in (4b).

(4) a. Ayuko-no ookii e hon
Ayuko-GEN big picture book
'Ayuko's big picture book' (Japanese)

b. Myungwoniui kun kurim chaek
Myungwoni-GEN big picture book
'Myungwoni's big picture book' (Korean)

Similarly, the rules building oblique noun phrases, i.e., noun phrases with postpositions that serve as oblique arguments of verbs, were ported directly. An example in Japanese is shown in (5a) with the corresponding Korean phrase in (5b).

(5) a. ooki ie ni
big house in
'in the big house' (Japanese)

b. kun jib eh
big house in
'in the big house' (Korean)

A rule fragment for these oblique noun phrases from the Japanese grammar is shown in (6). For illustrative purposes, we have shown a very simple rule; the annotations on most rules are significantly more complicated, which is why grammar porting is so desirable for bootstrapping grammar development.

(6) NPobl ⟶
{ Nadj: (^ OBJ)=!
PPobl: ^ =!
|AN: (^ OBJ)=!
PPobl: ^ =!
(!CHECK POST-TYPE)=c 'to-ni'
| ...}.

In the first disjunct in Rule (6) (from { to |), the NPobl consists of an Nadj which is the OBJ of the corresponding f-structure (Nadj: (^ OBJ)=!) followed by a PPobl postposition which is the head of the corresponding f-structure (PPobl: ^ =!).[2] The second disjunct (from | to |) is similar except that the PPobl is restricted to postpositions of the type `to-ni` and the OBJ of this postposition is an AN instead of the usual Nadj. Note that `to-ni` is a purely formal symbol whose spelling echos the surface forms of Japanese. The corresponding Korean form can be marked with this value to satisfy the constraint in this rule, or its spelling can be changed to make it more suggestive of the Korean realization. Other disjuncts are found in this rule, indicated here by |...}.

Other parts of the grammar that could be ported without change include the implementation of pro-drop for subjects and objects. Examples of sentences with pro-dropped subjects for Japanese and Korean are seen in (7). The analysis corresponding to the Korean sentence is shown in Figures 5 and 6.

(7) a. jitensha de ie ni kaeru.
bicycle by home to return
'(I/You/He/She/We/They) return home by bicycle.' (Japanese)

b. jajungu ro jib eh dorakanda.
bicycle by home to return
'(I/You/He/She/We/They) return home by bicycle.' (Korean)

Pro-drop is analyzed by optionally providing a null pronominal subject and/or object for each verb frame that subcategorizes for these functions. If an overt subject or object is found in the clause, then the pro-drop option is not chosen because the PRED of the overt subject would fail to unify with the PRED of the optional pronominal subject. However, if there is no overt subject, then the pro-drop option must be chosen because otherwise the subcategorization requirements of the verb would not be met. The null-anaphor template NA that provides the pronominal arguments is shown in (8a), where GF is a grammatical function value that is passed in by the verbal template. (8b) shows the expansion of the template for pro-dropped SUBJs.

(8) a. NA(GF) =
@(DEFAULT
(^ GF PRED) (^ GF PRON-TYPE) 'pro' null).

b. (^ SUBJ PRED)='pro'
(^ SUBJ PRON-TYPE)=null

[2] In the XLE grammar development platform, the ^ corresponds to the traditional LFG ↑ and the ! corresponds to the traditional LFG ↓.

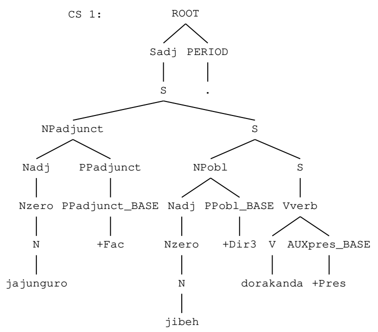


Figure 5: Pro-drop: Korean c-structure for (7b)

```
"jajunguro jibeh dorakanda."

  [PRED      'dorakada<[15-SUBJ:pro], [9:eh]>'
  |SUBJ      [PRED      'pro'
  |           PRON-TYPE null]
  |          [PRED      'eh<[8:jib]>'
  |           OBJ        [PRED 'jib'
  |OBL                  8[LOCATION-TYPE general, PERS 3]
  |           CHECK      [POST-TYPE ni]
  |          9[POST-FORM eh]
  |            { [PRED  'ro<[0:jajungu]>'
  |ADJUNCT     {  OBJ   [PRED 'jajungu'
  |            {        0[PERS 3]
  |            {  CHECK [POST-TYPE de]
  |            { 1[ADJUNCT-TYPE postpositional, POST-FORM ro] }
  |TNS-ASP   [MOOD indicative, TENSE pres]
15[PASSIVE -, STMT-TYPE decl, VTYPE main]
```

Figure 6: Pro-drop: Korean f-structure for (7b)

The ability to drop postpositional case and discourse function markers was also ported directly. In a standard SOV (or OSV) sentence, if the accusative case marker is dropped, but the nominative case marker is not, as in (9b), the sentence is given only one parse with the caseless noun phrase taking the object function and the nominative case marked noun phrase being the subject of the sentence. The same holds when only the nominative case marker is dropped but the accusative is not, as in (9c). When both case markings are dropped, as in (9d), the sentence is given two parses with each noun phrase being the subject in one parse and the object in the other. The Japanese equivalents of (9) receive the same analyses.

(9) a. Minjaga Taesunul boattda.
Minja-NOM Taesun-ACC saw
'Minja saw Taesun.' (Korean)

b. Minjaga Taesun boattda.

c. Minja Taesunul boattda.

d. Minja Taesun boattda.
'Minja saw Taesun.' (preferred due to default word order)

2.2 Modification of Rules

Although the typological similarities between the languages are striking, there are a few places where the annotated phrase structure rules had to be altered. The most important of these was in the analysis of sentential negation. Both Korean and Japanese have a type of affixal negation, shown in (10). However, Korean can encode negation as an adverb, similar to English, as in (11) (Kim, 2000). This construction is not found in Japanese and hence had to be added to the Korean grammar as part of the port.

(10) a. Taro ga modoranai.
Taro NOM return-NEG
'Taro isn't returning.' (Japanese)

b. Minja-nun doraka jianihanda.
Minja-TOP return-PRE -NEG
'Minja isn't returning.' (Korean)

(11) Minja-nun ani dorakanda.
Minja-TOP NEG return-PRES
'Minja isn't returning.' (Korean)

The structure for the Korean adverbial negation in (11) is shown in Figures 7 and 8. The annotated phrase-structure rules had to be modified to allow for an optional ADVneg in initial position in Vverb. The corresponding affixal negation construction in Japanese (10a) is shown in Figures 9 and 10 (Korean affixal negation in (10b) is identical). Note that although the c-structures differ between the languages, the f-structures are very similar with negation being an ADJUNCT to the main predicate and having ADJUNCT-TYPE `neg`.

Other syntactic differences between Korean and Japanese include the existence of double accusative constructions in Korean, as in (12), but not in Japanese.[3] This difference did not require any changes to the annotated c-structure rules since the restriction against double accusatives

[3] In contrast, double nominative constructions are found in both languages and hence were ported directly.

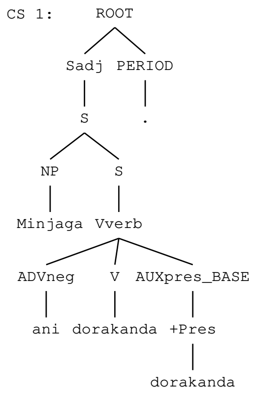


Figure 7: Adverbial negation: Korean c-structure

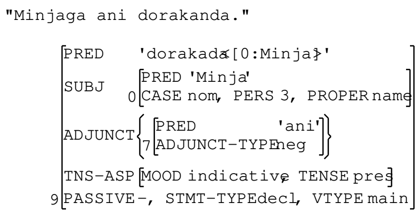


Figure 8: Adverbial negation: Korean f-structure

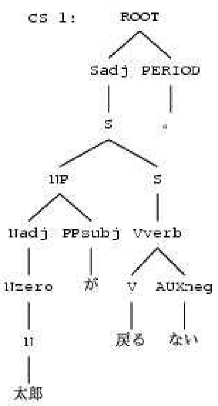


Figure 9: Affixal negation: Japanese c-structure

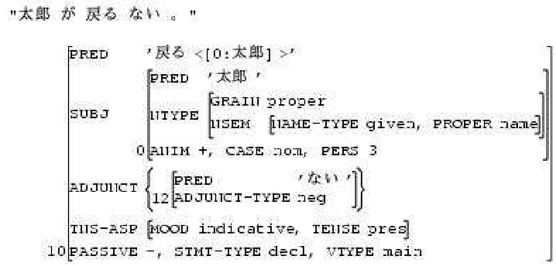


Figure 10: Affixal negation: Japanese f-structure

is not done on the phrase structure but rather in the subcategorization frames of the relevant verbs (see (O'Grady, 2002) for a detailed analysis of this construction).

(12) Kay-ka haksayng-ul tali-lul mwul-ess-ta.
dog-NOM student-ACC leg-ACC bit
'The dog bit the student on the leg.' (Korean)

In addition to the double accusative construction in (12) in which both accusatives are nominal arguments of the verb, there is another type involving complex predication, as in (13) (see (Lee, 1993) on complex predication in Korean).

(13) Minjaga badakul chungsorul hattda.
Minja-NOM floor-ACC clean-ACC do
'Minja cleaned the floor.' (Korean)

These are similar to the *suru* complex predicates found in Japanese except for the case marking. As such, we hope to extend the Japanese complex predicate analysis to Korean with only minor modification of the phrase structure rules to allow case marking within the complex predicate.

Finally, there are some differences between the languages in the nominal classifier system and in the multiple marking of quantification;[4] we have yet to explore these.

3 Porting Lexicons and Morphologies

Unlike the annotated c-structure rules, the lexical items differ significantly between Japanese and Korean. However, once the lexical item head

[4] We would like to thank an anonymous reviewer for bringing the multiple quantification to our attention.

words are changed, the information for many entries can remain the same. For example, the entry for the Japanese accusative postposition *o* is identical to that of the Korean accusative postposition *ul/rul* other than the form of the postposition, e.g., both assign accusative case. This was similar for the majority of closed and open class items. Thus, by using a Japanese to Korean dictionary to translate the head words in the lexicon, a detailed lexicon can be semi-automatically created for Korean.[5] However, at this point in our experimentation, we are still working with a small lexicon for open class items, although all the closed class items have been translated.

In addition to the lexicon, the tokenizer and morphology are done differently in the Japanese and Korean grammars. The Japanese grammar uses the ChaSen tokenizer (Asahara and Matsumoto, 2000) to insert token boundaries and determine certain part of speech information. Since the Korean writing system puts spaces between words, similar to English, a tokenizer was ported from the English grammar. The tokenizer inserts token boundaries between the space-delimited words and around punctuation. The resulting tokens are then fed into a finite-state morphology (FSM) for Korean (Beesley and Karttunen, 2003). In the current version of the grammar, this FSM works on Latin transliterated Korean like the rest of the grammar, but existing Korean FSMs can easily be incorporated into the grammar. Despite these differences between the Japanese and Korean morphologies, the same annotated phrase structure rules could be used. For example, as seen in Figure 3, the Japanese case markers *ga* and *o* are treated as separated words of category PPsubj and PPobj. Figure 1 shows that the Korean case markers are morphological tags +Nom and +Acc, but theses tags are of category PPsubj and PPobj and hence interact with syntactic rules just as in Japanese.

In addition to these lexicon and tokenization and morphology preprocessing steps, some minor changes to the core annotated phrase structure rules were needed in the domain of suffix syntax. For example, the Japanese grammar allows both orders for the location suffix in conjunction with the topic suffix. However, in Korean, the only possible order is for the location suffix to be followed by the topic suffix (e.g., *eh nun*); thus, the rule had to be further constrained for the Korean grammar.

4 Conclusion

We are encouraged by our success in this preliminary investigation into using an existing deep grammar to develop a new one for a typologically similar language. With only two person-months of effort, we found that major parts of the Japanese LFG grammar could be carried over unchanged into the Korean grammar. Most of the core annotated phrase structure rules remain the same, and it seems that many lexical items can be ported merely by changing the head-word of the Japanese entry to its Korean equivalent. New finite-state machines for tokenization and morphological analysis had to be created and incorporated into the system, as was to be expected.

Much work remains to be done to bring Korean coverage and accuracy up to the level of the Japanese and other ParGram grammars. This work will focus on peripheral syntactic rules and expansions to both the lexicon and morphology, with relatively little modification anticipated for the rules of major syntactic constructions. Based on our current rate of progress, we estimate that the Korean grammar will reach a level comparable to the current Japanese with a total of eight months of effort, about a third of the time we would have expected it to take to develop a Korean grammar from scratch. Of course, a final assessment of grammar-porting effectiveness will eventually also require systematic, corpus-based evaluations of coverage and accuracy.

We conclude from this limited experiment that porting LFG grammars across typologically similar languages is an effective method for bootstrapping multilingual grammar development. Outside the domain of LFG grammar development, our experience suggests that grammars written in other formalisms can also be used in similar grammar ports. For example, the Japanese HPSG grammar (Siegel and Bender, 2002) should lend itself to the rapid creation of a Korean grammar, especially if it exploits the concepts developed in the gram-

[5]There has been significant work on Korean/Japanese machine translation, e.g. (Paik and Shirai, 2001), including the development of lexical resources (Paik et al., 2001).

mar MATRIX project (Bender et al., 2002). Thus, we hope that the techniques described here will be exploited more generally in the development of broad-coverage deep grammars for a range of languages.

References

Masayuki Asahara and Yuji Matsumoto. 2000. Extended models and tools for high-performance part-of-speech tagger. In *Proceedings of International Conference on Computational Linguistics (COLING)*, pages 21–27.

Kenneth Beesley and Lauri Karttunen. 2003. *Finite-State Morphology: Xerox Tools and Techniques*. CSLI Publications, Stanford, California.

Emily Bender, Dan Flickinger, and Stephan Oepen. 2002. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the Workshop on Grammar Engineering and Evaluation*, pages 8–14. COLING 2002 workshop.

Miriam Butt, Tracy Holloway King, Maria-Eugenia Niño, and Frédérique Segond. 1999. *A Grammar Writer's Cookbook*. CSLI Publications.

Miriam Butt, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. The parallel grammar project. In *COLING 2002: Workshop on Grammar Engineering and Evaluation*, pages 1–7.

Mary Dalrymple. 2001. *Lexical Functional Grammar*. Academic Press, New York. Syntax and Semantics, volume 34.

Anette Frank. 1999. From parallel grammar development towards machine translation. In *Proceedings of MT Summit VII*, pages 134–142.

Ronald Kaplan and Joan Bresnan. 1982. Lexical-functional grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281. The MIT Press.

Jong Bok Kim. 2000. *The Grammar of Negation: A Constraint-based Approach*. CSLI Publications, Stanford, California.

Sookhee Lee. 1993. The syntax of serialization in Korean. In Patricia Clancy, editor, *Japanese/Korean Linguistics*, volume 2, pages 447–463. CSLI Publications, Stanford, California.

Hiroshi Masuichi and Tomoko Ohkuma. 2003. Constructing a practical Japanese parser based on Lexical-Functional Grammar. *Journal of Natural Language Processing*, 10(2):79–109. To appear; in Japanese.

John T. Maxwell, III and Ronald M. Kaplan. 1993. The interface between phrasal and functional constraints. *Computational Linguistics*, 19:571–589.

William O'Grady. 2002. Korean case: A computational approach. *Korean Linguistics*, 11:29–51.

Kyonghee Paik and Satoshi Shirai. 2001. Exploiting linguistic similarities for machine translation: A case study of Japanese-to-Korean. In *Proceedings of International Conference on Speech Processing*, pages 737–742.

Kyonghee Paik, Francis Bond, and Satoshi Shirai. 2001. Using multiple pivots to align Korean and Japanese lexical resources. In *NLPRS-2001*, pages 63–70.

Stefan Riezler, Tracy Holloway King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell III, and Mark Johnson. 2002. Parsing the wall street journal using a lexical-functional grammar and discriminative estimation techniques. In *Proceedings of the ACL*.

Melanie Siegel and Emily Bender. 2002. Efficient deep parsing of Japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization*, pages 31–38. COLING 2002 workshop.

Ho-min Sohn. 1999. *The Korean Language*. Cambrige University Press, Cambridge. Chapter 2.4.

Using a MetaGrammar for parallel multilingual grammar development and documentation

Alexandra Kinyon
Department of Computer Science
University of Pennsylvania
kinyon@linc.cis.upenn.edu

Owen Rambow
Department of Computer Science
Columbia University
rambow@cs.columbia.edu

Abstract

In this paper, we build up on the work presented in (Kinyon and Rambow, 2003) and (Clément and Kinyon, 2003b). We describe how a MetaGrammar tool allows us to automatically generate, from a hand-crafted hierarchy, multilingual LFGs for English, French and German, as well as test-suite to document the coverage of our grammars, and help us detect errors. We explicitly maximize the sharing of constraints at the metagrammatical level. We also discuss the extension of this cross-language sharing to "less related" languages, such as Russian.

1 Introduction

Few multilingual parallel grammars exist and when they do, developing and maintaining such multilingual grammars is a challenging task. (Butt et al., 2002) report on the development of parallel LFGs for French, English, German, Japanese, Norwegian and Urdu and explain that for each language, a grammar is crafted at a different site. Consistency issues are solved during periodic meetings in which, when possible, participants agree on "sharing" a linguistic analysis across languages. But, to the best of our knowledge, there is no explicit cross-language rule sharing in that those grammars, even if they encode a consensual syntactic theory, are physically separate. In this paper, we explore the use of a higher level of syntactic abstraction called the MetaGrammar (MG) to automatically generate, from a single hand-coded hierarchy, multi-lingual LFGs for French, English and German, as well as annotated sentences to illustrate the grammar coverage and help detect errors. The main advantage of automatically generating grammars from a MG are a reduced need for human intervention in the grammar development process, the explicit sharing of rules for two or more languages at the MetaGrammatical level, and an increased consistency and coherence in the grammars generated.

In section 2, we introduce the notion MetaGrammar. In section 3, we present the MG tool we use and we explain how we generate from a single compact MG hierarchy LFGs with an explicit sharing of rules across languages. We also explain why we generate - from that same single hierarchy - multilingual annotated sentences, and how it helps us document the grammar and detect errors. In section 4 we explain the advantages of our MetaGrammar approach. In section 5, we focus on specific syntactic phenomena (Wh-extraction, passive, word order variations for ditransitives) to illustrate the cross-language rule sharing mechanism. In section 6, we discuss the extension of our approach to "less related" languages. More specifically, we take the example of Russian, which has a rich morphology and a relatively free word order. Finally, in section 7, we contrast our approach with other approaches such as Matrix grammars.

Note that this is work in progress, and we do not intend to compete with the large effort put in the ParGram project, but simply resort to LFG to illustrate the advantages of a grammar engineering technique which has successfully been used to develop a wide-coverage Tree Adjoining Grammar (TAG) for French (Abeillé et al., 1999), English (Xia, 2001), a medium-size TAG for Italian (Candito, 1999), German (Gerdes, 2002). These grammar development efforts were not unified (i.e. no attempt was made to craft a unique MG hierarchy for several languages, and furthermore these grammars were developed independently, using distinct - and often incompatible- MetaGrammar implementations). Here, we attempt unify some of these cross-linguistic efforts into a single MG hierarchy. Also, as developed in (Kinyon, 2003), (Clément and Kinyon, 2003b) and (Kinyon and Rambow, 2003), by producing LFG (and not TAG) we emphasize the fact that the MG is not framework-dependent, and to a large extent portable to most syntactic frameworks.[1]

2 What is a MetaGrammar ?

The notion of MetaGrammar was originally presented in (Candito, 1996) to automatically generate a TAG with 5000 elementary trees[2] (Abeillé et al., 1999). Her goal was to organize framework-independent syntactic knowledge in a compact 3-dimensional hierarchy from which TAG grammars

[1]Here, we focuss on multilingual sharing. The scalability of a MG approach has been demonstrated in the work on TAG cited above, since the TAGs generated contain several thousand rules.

[2]We do not present TAG here. Suffice it to say that the basic rules are phrase-structure trees.

could be automatically generated[3]:

- Dimension 1: Initial subcategorization
- Dimension 2: Valency alternations
- Dimension 3: Surface Realization of arguments

Related to the notion of MetaGrammar, (Kinyon, 2000) introduced the notion of *HyperTag*. The main idea behind HyperTags is to keep track, when trees (i.e. grammar rules) are generated from a MetaGrammar hierarchy, of which terminal classes were used for generating the tree. This allows one to obtain a framework-independent feature structure encoding the salient syntactic characteristics of each rule.[4] For instance, the verb *give* in *A book was given to May* could be assigned the HyperTag:

$$\begin{bmatrix} \text{Subcat} & \text{Ditransitive} \\ \text{Valency alternations} & \text{Passive no Agent} \\ \text{Argument Realization} & \begin{bmatrix} \text{Subject:} & \text{Canonical NP} \\ \text{Object:} & \text{Not realized} \\ \text{By-Phrase:} & \text{Canonical PP} \end{bmatrix} \end{bmatrix}$$

3 Generating LFGs

In order to generate LFGs, we retain the linguistic insights of (Candito, 1996) - namely the three-dimensional hierarchy - but reformulate it slightly: dimensions one and two are unchanged and encode respectively "initial subcategorization" and "valency alternations". Dimension three is reformulated to encode not only the surface realization of arguments, but also the surface realization of heads (ex: verbs), and modifiers. We also add a fourth dimension for lexicalization i.e. to "plug in" the lexical items which will appear in the annotated sentences of our test-suites, as well as a fifth dimension for sentence type (e.g. main clause with or without a subordinate ...).

We do not use Candito's implementation. Instead, we use a newer and more framework independent MG tool, developed at LORIA, and presented in (Gaiffe et al., 2002).[5]

In the LORIA tool, each class in the MG hierarchy encodes:

- Its SuperClass(es).
- A Feature structure (i.e. HyperTag) which captures the salient linguistic characteristics of that class.
- Which resources the class needs from other classes and provides to other classes.
- The topological content of the class, consisting of:
  - A set of quasi-nodes (i.e. variables).
  - Relations between these nodes (*parent, dominates, precedes, equals*).
  - A function for each quasi-nodes to decorate the tree (e.g. agreement equations and/or LFG functional equations).

A class inherits its HyperTag, resource needs and supply lists, and topological content (TC) by unifying its parents' corresponding items. New specifications can be added in the definition of the class. The MG tool automatically crosses the nodes in the hierarchy, i.e., it creates new classes that inherit from existing classes in the hierarchy. These new classes must be "balanced" classes, that is classes that do not need nor provide any resources. Then for each balanced terminal class, $<$HyperTag, tree$>$ pairs are generated. Since the LORIA tool was originally developed to generate TAGs, $tree$ was then interpreted as a TAG elementary tree i.e. grammar rule. We extend the approach, building up on the work of (Kinyon and Rambow, 2003) and (Clément and Kinyon, 2003b): $tree$ is now a phrase structure tree, including lexical items, and is decorated with LFG functional equations.

So the grammar generation process takes place in two steps. In the first step, we manually encode a MG hierarchy. The classes in the hierarchy encode nodes which are decorated with standard LFG equations (including "functional uncertainty" (Kaplan and Zaenen, 1989) for long-distance dependencies). From this MG hierarchy, the tool automatically generates trees annotated for phrase structure (as explained in (Kinyon and Rambow, 2003)). This allows us to obtain a multilingual test-suite. These generated trees are decorated with LFG functional information, (as was first proposed by (Kameyama, 1986), and in a way similar to the work of (Frank, 2000)). In a second step, these decorated tree are decomposed into standard LFG rewriting rules as well as LFG lexical entries, thus yielding a grammar.

Figure 1 illustrates the generation of a decorated tree for a simple English sentences ("*Peter sees Mary*"), and the decomposition process into two LFG rewriting rules and three lexical entries. Note that with this simple example, the three classes apply to all three languages: French, German, English (modulo the lexical item "see" plugged in CLASSTRANSITIVE for sake of brevity)[6]. The rules can then be interfaced with a parser[7].

[3]It is also worth mentioning that (Candito, 1999) explored whether some components of her hierarchy could be re-used across languages (French and Italian). However, she developed two distinct hierarchies for these languages, and generated only TAGs.

[4]The notion of HyperTag was inspired by that of supertags presented in (Srinivas, 1997), which consists in assigning a TAG elementary tree to lexical items, thus enriching traditional POS tagging. However, HyperTags are framework-independent.

[5]This compiler is freely available on http://www.loria.fr/equipes/led/outils/mgc/mgc.html

[6]The figure is simplified for space, esp. all morphological features are omitted and in the "real" hierarchy, lexical items are "plugged in" in a language-specific dimension 4.

[7]We use the freely available XLFG parser described in

In addition to yielding LFG rules and lexical entries, the decorated tree constitutes an annotated sentence i.e. all the decorated trees we generate constitute a test-suite, and this test-suite is the kernel from which our grammars our generated. As detailed in (Kinyon and Rambow, 2003), the main difference between our test-suite generation and "standard" test-suites is that the latter are generally created by first choosing a set of sentences, which are then automatically parsed and whose annotations are manually disambiguated and/or corrected, whereas our test-suites are entirely generated from the MG hierarchy, which ensures a better annotation consistency: if an error is detected in our test-suite, it is corrected in the MG hierarchy and never in a post generation phase. The test-suite is useful as a documentation tool to illustrate which phenomena are handled by the grammar, and it also allows for detecting errors in the grammar: if some of the generated sentences are ungrammatical, or the annotation deemed incorrect, this means that some grammar rules are wrong and changes have to be made in the MG hierarchy. [8]

We do not detail here how LFG, as a framework, benefits from the use of MetaGrammar[9], however, note that The "resource model" of the MetaGrammar, based on "needs" and "provides", allows for a natural encoding and enforcement of LFG coherence, completeness and uniqueness principles: A transitive verb needs exactly one resource "Subject" and one resource "Object". Violations result in invalid classes which do not yield any rules. So in that perspective, it makes little sense, apart from practical reasons such as interfacing the grammar with an existing parser, to force the rules generated to be trees of depth one. Also, a similar notion of resource-sensitivity is used within the LFG community, e.g. to compute semantic forms using linear logic (Dalrymple et al., 1995), or to account for discontinuous constituents such as the German split NP construction (Kuhn, 1998) (*Ex: Bücher sieht Anna drei - lit: Books sees Anna three*). As explained in (Joshi and Vijay-Shanker, 1989), one major difference between a framework such as TAG and LFG is that TAG exhibits an extended domain of locality.[10] In addition to being framework-independent, the MG does not impose a given domain of locality. Here, we have chosen the sentence as our domain of locality[11]. This implies that discontinuous constituents, such as the split NP construction for German, or its French equivalent (*"Des livres Anna en a vu trois"* lit: Books Anna them:clitic has seen three) pose no major problem to encode in the MG hierarchy. Similarly, modifier extraction from internal arguments are easily handled in the MG hierarchy *(ex: Which castle did John take a picture of)*. Problems may be encountered though, based on the limitation of the framework one chooses to generate from the MG. So, when generating LFGs, the split NP construction and its French equivalent with the addition of the clitic, will be problematic w.r.t. well-formedness conditions of F-structure: "en" and "livres" are both realizations of the direct object, and can be considered as a violation of the uniqueness constraint. Similarly, for modifier extraction, no problem will be encountered when generating an LFG from the MG. However, the generation of a TAG would be problematic (at least when using the standard predicate argument co-occurence principles, which explicitly excludes modifiers from the domain of locality of a grammar rule).

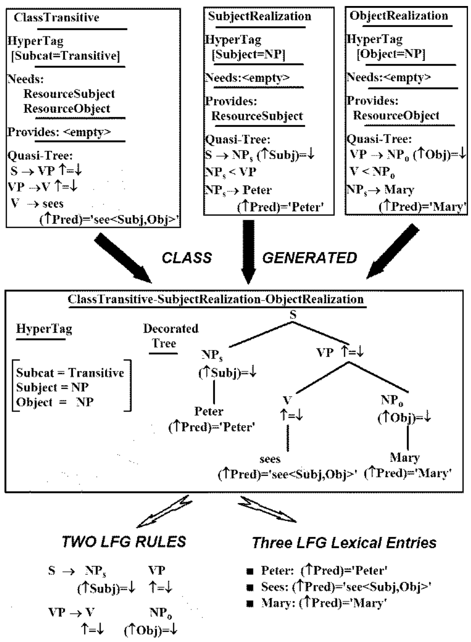


Figure 1: Generation of a decorated tree which yield two LFG rules and three lexical entries
$\rightarrow$ stands for father, $<$ for precedes. Standard $\downarrow$ $\uparrow$ notations are used for LFG.

(Clément and Kinyon, 2001), and have experimented with the Medley LFG parser described in (Kaplan and Maxwell, 1996).

[8] Of course, this strategy does not detect all errors and the documentation value of the test-suite is complementary to additional documentation of the grammar development process but does not claim to replace it.

[9] This point is discussed in detail in (Clément and Kinyon, 2003b) and (Clément and Kinyon, 2003a)

[10] Another way to say this is that TAG rules are trees of depth greater than one, whereas LFG rules may be viewed as trees of depth one, with the root of the tree being the left hand-side of a rule, and the children being the right hand-side of a rule.

[11] By contrast (Clément and Kinyon, 2003b) generate LFGs but use the same domain of locality of TAG i.e. clauses.

4 Advantages of the MetaGrammar

A first advantage of using a MetaGrammar, independently of which grammar framework one chooses to generate using the MetaGrammar, is, as argued in (Kinyon and Prolo, 2002), that the syntactic phenomena covered are quite systematic: if rules are generated for “transitive-passive-whExtractedByPhrase” (e.g. *By whom was the mouse eaten*), and if the hierarchy includes ditransitive verbs, then the automatic crossing of phenomena ensures that sentences will be generated for “ditransitive-passive-whExtractedByPhrase” (i.e. *By whom was Peter given a present*). All rules for word order variations are automatically generated by under-specifying relations between quasi-nodes in the MetaGrammar hierarchy (e.g. precedence relation between first and second object for ditransitives in French). A second advantage of the MG is to minimize the need for human intervention in the grammar development process: The grammar developer encodes the linguistic knowledge in a compact manner i.e. the MG hierarchy, and then verifies the validity of the rules generated. If some grammar rules are missing or incorrect, then changes are made directly in the MG hierarchy and never in the generated rules. This ensures a homogeneity not necessarily present with traditional hand-crafted grammars. A third and essential advantage is that it is straightforward to obtain from a single hierarchy parallel multi-lingual grammars such as the parallel LFG grammars presented in (Butt et al., 1999) and (Butt et al., 2002), but with an explicit sharing of classes. Finally, we have mentioned that we generate a test-suite in parallel to generating a grammar.

5 Sharing classes across languages

There are two ways in which MetaGrammars for two languages can differ: in the class hierarchy, and in the topological content (TC). A difference in class hierarchy reflects the availability of a high-level grammatical phenomenon in one language, but not the other, such as clitics for French, which do not exist in German or English. A difference in TC reflects a difference in the way in which high-level phenomena affect the structure in the word order. Thus, in general, TC differences are much more common than differences in the class hierarchy. Note that often it is a matter of choice whether a difference between two languages is represented as a difference in the hierarchy, or a difference in the TC. We have implemented a hierarchy encoding a fragment of the syntax of English, French, and German. From this single hierarchy, grammars (and test-suites) are automatically generated. Language-specificity is encoded in the HyperTags of classes in the MG hierarchy as features such as [German:+/-]. This prevent unification, and thus the crossing, of incompatible classes. For instance, a class like TRANSITIVEWITHPARTICLE-CLITICIZEDOBJECT is not created because the class TRANSITIVEWITHPARTICLE contains in its HyperTag [French=-; German=+; English=+], and the class CLITICIZEDOBJECT contains [French=+;German=-;English=-], making the two HyperTags non unifiable. Therefore, in the discussion below, we assume that in deriving a rule of a grammar, only classes and TCs with compatible linguistic features are used.[12]

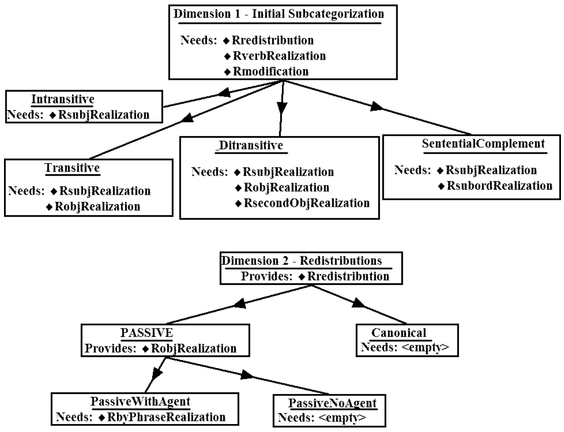


Figure 2: Fragments of dimensions 1 and 2 shared by all 3 languages (resources start with ”R” by convention)

We choose to maximize the number of shared classes from the hierarchy. Thus, the basic classes from the first, second and third dimensions are the same in all three languages. All three languages have various subcategorization frames, all three languages have passive, with or without agent. Figure 2 illustrates some of the classes shared by all three languages in dimensions 1 and 2. It is simplified for readability, the TC and HyperTags of the classes are omitted.

Many classes in dimension 1 are shared by the three languages: all three have intransitive, transitive, ditransitive verbs, verbs which take sentential complements, and verbs which take an indirect object introduced by a preposition which is triggered by the verb, such as:

- G: Das hängt vom Kontext ab (lit: it depends from context:dative on:particle)

[12]One may want to introduce a hierarchical notion of linguistic features, so that [German:+] may be compatible with [Germanic:+] to capture commonalities in the Germanic languages. It is unclear how productive this would be; possibly only at a lower lever (i.e., Ripuarian German shares much with Palatinate German, but less with Bavarian German).

- G: Es besteht aus/*von vielen Teilen (lit: it consists of many parts:dative)
- E: It depends on/*of the context
- E: It consists of/*in many parts
- F: Cela dépend de/*à la situation (lit: it depends on the situation)
- F: Cela correspond à/*de notre point de vue (lit: it corresponds to our point of view)

However, some classes in dimension 1 can be language specific. For instance TRANSITIVEWITHPARTICLE (ex: *I would like to pick a new computer up*) is a valid subcategorization frame for both German and English, but not for French[13]. Also, some classes in dimension 2 are language specific. For instance, the class PASSIVE has a subclass IMPERSONALPASSIVE valid for German (ex: *Es wurde getanzt / lit: it was danced*), but not for English. The acceptability of impersonal passives for French is debatable (? Il a été dansé) and its occurrence at most marginal, so we have also ruled it out.[14] Note that we handle "English dative shift" in dimension 3, and not in dimension 2. The main motivation is that "dative-shift", unlike passive, does not trigger a change of the syntactic function of the arguments of the verb, but only a change in phrase-structure realization and in word order. In *John gives a book to Mary* versus *John gives Mary a book*, first it is difficult to assess which of the two should be considered as canonical. Second, for both realizations, "book" is a direct-object and "Mary" a second-object, regardless of whether the second object is realized as a NP or a PP. Moreover, not all ditransitives may undergo dative-shift in English. In our hierarchy, we have three distinct subclasses of DITRANSITIVE in dimension 1 - one for "dative-shiftable" ditransitives such as "give", which is specific to English - one for ditransitives which allow only for an NP PP construction (ex: *He submitted a paper to the workshop / *He submitted the workshop a paper*), which is valid for English and French - one for verbs which allow only for the double NP construction (ex: *He asked me a question /*He asked a question to his relatives.*, which is valid for English and German.

In dimension 2, the class PASSIVE also has a subclass DATIVEPASSIVE, further subdivided into WITHAGENT and WITHOUTAGENT, which is valid only for English and German, but not for French. The DATIVEPASSIVE triggers the use of the auxiliary "bekommen" in German, whereas the STANDARDPASSIVE triggers the use of the auxiliary "werden". This accounts for the data:

- English:
  - *A book was given to Mary.*
  - *Mary was given a book.*
  - **To Mary was given a book.*
- French:
  - *Un livre a été donné à Marie.*
  - **Marie a été donné un livre.*
  - Rmk: *A Marie a été donné un livre.* is grammatical, but it is a subject-inversion, not a dative passive.
- German:
  - *Ein Buch wurde dem Mann gegeben*
  - *Dem Mann wurde ein Buch gegeben* (lit: the man:dat was a book:nom given) i.e. not dative-passive but subject-inversion
  - **Der Mann wurde ein Buch gegeben* (lit: the man:nom was a book given)
  - *Der Mann bekommt ein Buch / hat ein Buch bekommen* (lit: The man:nom. is given a book:acc/ was a book given).[15]

Moreover, all three languages allow *wh*-movement for all arguments. Furthermore, the TC of *wh*-movement is the same: the moved element must end up in a position c-commanding its original position, subject to path constraints.

To summarize, for canonical ditransitives, the realizations for the three languages are:

- English: dative-shiftable verbs (give NPo PP/ NP2 NPo), double NP only (ask) or NP PP only (submit), with fixed word order.
- German: double NP (geben NP:dat NP:acc)
- French: NP PP, with unconstrained order between the two.

Where the languages differ is most saliently in word order, which is encoded in dimension 3.[16] A general observation is that for all the dimensions, classes higher in the hierarchy are more likely to be shared across-languages, whereas leaves are more

[13] Of course the particle will vary depending not only on the verb (call up / blow off), but also on the language, but this is encoded by having a RPARTICULEREALIZATION resource, which is provided in dimensions 3 and 4 in a language-specific manner.

[14] As anecdotal evidence, "Es wurde getanzt" yielded 932 google hits. By contrast, "Il a été dansé" yielded 10 hits, all of which pertaining to the discussion of impersonal passive.

[15] Note that in the active form, e.g. *Ich gebe dem Mann ein Buch*, the second object is dative. It becomes nominative when promoted to subject.

[16] We do not provide a figure for dimension 3, which is a multiple inheritance network, for space reasons.

likely to be language-specific. Also, classes in dimensions 1 and 2 (subcat. and valency alternations), are more likely to be shared than classes from dimension 3, because the latter encodes word-order variations i.e. classes in dimension 3 have a richer TC.

First, the German verb-final clause is implemented by having language-specific TC statements on the classes for the direct and indirect objects (which supply the resources required by transitive and intransitive verbs). For English and French, these NP nodes are ordered to the right of the lexical verb, while in German, they are ordered to the left of the position from which the clause is projected.

Second, we must account for the relatively free modifier insertion in German and French, but not in English.

- E: *The boy sees today a duck
- G: Der Junge sieht heute eine Ente
- F: Le garcon voit aujourd'hui un canard

We do so by so supplying linear precedence (LP) rules in the TC statements for phrase-structure realization in these languages. E.g., for English ditransitives, the direct object NP must precede the indirect-object PP, while in German and French, the order of the two objects is unconstrained. Furthermore, in German, there are no LP restrictions on the subject, either, while in French (as in English), it is ordered to the left of the main verb. Thus, note that word order variation among all arguments in German, and among the two objects in French, is not represented as a class in the hierarchy, but as underspecification in TC statements.

Finally, we must deal with the verb-second effect in German. We do so by introducing a new class into the hierarchy, which supplies a resource V2 which is required by all non-complementizer clauses. (All clauses for all three languages are divided into complementizer and non-complementizer clauses.) This class has a realization which establishes two positions at the beginning of the verbal projection, one obligatory one for a finite verb, and one obligatory one for an arbitrary element to its left (Vorfeld). The position for the finite verb blocks the free permutation of elements permitted to its right (otherwise the V2 would not be enforced). These two positions can be filled by extrinsic elements: an auxiliary for the verbal position, an adverbial or an expletive *es* for the Vorfeld. We also can fill the Vorfeld and V2 positions with material from the verbal structure itself: the verbal position can of course be filled by the main lexical verb (if it is finite), and the initial position can be filled by any argument (through the usual mechanism of *wh*-movement).

Note that we are free to associate classes with HyperTag content. The V2 class regulates word order but is not optional – it is obligatorily used when forming a non-complementizer clause. Thus, it does not contribute to the HyperTag. On the other hand, a class such as PASSIVENOAGENT is entirely optional, and so, when it is used, it should contribute to the HyperTag.

Exceptionality is handled essentially in the lexicon and not so much in the MG hierarchy[17] e.g. when a transitive verb does not passivize, it is marked in its lexical entry (ex: French "posséder" -lit:possess-own). In any event, a MG organization does not increase the burden of exceptionality compared to traditional hand-crafted grammars.

6 Extension to less related languages

The languages we have examined (French, English and German) are quite-related i.e. all three are Indo-European languages. However, they are also quite distinct, since French is Romance language, while English and German are Germanic languages. As we extend our approach to more distinct languages, the less related the languages are, the fewer classes one expects to be able to share in the MG hierarchy. Since we are not going to develop grammars for all languages, the best educated guess we can make is that there is a continuum in "sharing". If we take the example of Russian, which is still an Indo-European language, but of the East-slavonic group, we may reuse for the most part the classes already developed for German, English and French. More specifically, we may retain our handling of Wh-extraction (1) and (2) below, of passive [18], of ditansitives (3). The major distinction is that we need to account for the richer morphology[19], and the free word order between the verb and its arguments by leaving precedence constraint under-specified in the TC of our class hierarchy: e.g. as in latin, all permutations between subject, object and verb are grammatical, as long as the case-marking allows one to correctly retrieve the correct syntactic function (e.g. 4, 5, 6, 7, 8, 9 below). The word order is essentially determined by pragmatics and by a topic-focus interpretation of the sentence.[20]

1. Каторую книгу она купила (lit: which book:acc she:nom bought)

[17] This is not specific to the MG, but simply stems from the fact that most "syntactic exceptions" are lexically determined.

[18] Although passive occurs less often in Russian than in fixed word-order languages, i.e. to express topic, Russian has a passive construction, with the "by-phrase" realized in the instrumental case with no preposition

[19] Russian has 6 cases: nom, acc, gen, dat, instrumental and locative.

[20] (8) and (9), with the verb in first position, would seem less frequent than the four other permutations, but grammatical.

2. Каторую книгу ты думаешь она купила (lit: Which book:acc you think she:nom bought)
3. Я подарил Маше книгу (lit: I:nom gave Macha:dat book:acc)
4. Книгу Лена купила (а журнал, она нашла) (lit: [The] book Lena bought, but [the] newspaper, she found)
5. Книгу купила Лена (lit: [The] book:acc bought Lena:nom)
6. Лена Книгу купила (lit: Lena:nom book:acc bought)
7. Лена купила книгу (lit: Lena:nom bought [a] book:acc)
8. Купила Лена книгу (lit: Bought Lena:nom book:acc)
9. Купила книгу Лена (lit: Bought Lena:nom book:acc)
10. Мать любит дотчь (lit: mother:nom/acc likes daughter:nom/acc)

Russian is considered SVO, because when the case-marking is morphologically ambiguous, as in (10), in a construction NP1 V NP2, NP1 is clearly understood as the subject, and NP2 as the object i.e. "The mother likes the daughter" and not "The daughter likes the mother".[21]

Examples of complex morphosyntacic phenomena, which would require particular attention for adding Russian to our MG hierarchy are:

The use of "numerals" in the nominative or accusative, followed by a noun (ex: three brothers)[22]:

- "one" is followed by a noun in the nom-sing (ex: Один брат - one brother)
- "two, three, four" are followed by a noun in the gen-singular (ex: два брата - two brothers)
- "five to 20" is followed by a noun in the genitive plural (ex:пять братьев - five brothers)
- "21" is followed by a noun in the nom-sing
- "22, 23, 24" + genitive singular
- "25 to 30" + genitive plural etc.

The morphological case of verb complements, which is not determined only by the verb, but also by the gender and animacy of the complement:

- Она видела книгу:acc-fem (lit: She saw book)
- Она видела журнал:acc-masc-inan, same declension as nom-masc (lit: She saw newspaper:acc)
- Она видела брата:gen-sing-masc-anim. (lit: She saw brother:gen)

Negation, which may trigger a change from accusative to genitive:

- У мења брат/*брата (I have a brother:acc/*gen)
- У мења нет брата/*брат (I have no brother:gen/*acc)
- Я не купила книгу/книги:acc-sing/gen-sing (lit:I didn't buy book:acc/gen) i.e. so called "case alternation")

Russian exhibits a complex verbal system w.r.t. aspect. All verbs come in a perfective/imperfective pair (+ a third option for movement verbs such as "go"), which would warrant a language-specific verb hierarchy in the MG.

An interesting point is that the "language class" does not always exactly reflect potential rule-sharing in practice. For instance, Russian has a reflexive clitic (ся), which behaves like the Italian "si" i.e. it agglutinates to the end of the verb (whereas French, which is more closely related to Italian, has no clitic agglutination).

It is likely, however, that if we were to develop a MG hierarchy for a non Indo-European language, fewer MG classes would be shared. For instance, if we were to add Japanese, we would have to develop a specific hierarchy for its complex particle system.

7 Comparison with related work

We have already mentioned the LFG ParGram project which aims at building parallel LFGs for a variety of languages, and have explained that the main difference with our MetaGrammar approach is that we explicitly resort to (and try to maximize) cross-language sharing of classes at the MG level.

Another approach which aims at building parallel grammars for English, German and Japanese is the Matrix approach described in (Bender et al., 2002). Matrix relies on HPSG type inheritance hierarchy to encode linguistic constraints (similarly to what is presented in (Flickinger, 1987)) to represent "cross-linguistic generalizations". It would seem that the main similarity between our MetaGrammar approach and the Matrix approach is that both resort to explicit cross-language sharing. The main difference between our MG approach and Matrix is that for Matrix the type-inheritance hierarchy is an inherent part of the formalism (i.e. HPSG), whereas our MG approach is a grammar engineering technique which allows one to generate grammars offline. Hence, a MetaGrammar hierarchy is

[21] It was recently brought to our attention that (King, 1993) argues in favor of Russian being VSO. We did not have time to investigate that point in more detail. It is, however, unlikely that it would affect the practical issues discussed here.

[22] When the numeral in another case i.e. gen, dat, instru., loc, the noun agrees in case with the numeral

framework-independent and not an inherent part of the frameworks of the grammars generated (which may be LFGs, TAGs and could conceivably be HPSGs).

8 Conclusion

We have presented a MetaGrammar tool which allows us to automatically generate from a single compact MG hierarchy LFG grammars for English, French and German as well as test-suites to document grammar rules and to help error detection. We have discussed the extension of this approach to a "less related language" such as Russian. We keep enriching our hierarchy in order to increase the coverage of our grammars, and plan to add new languages. However, hand-crafted grammars have intrinsic limitations, esp. when it comes to lexical coverage and the interface between the grammar and the lexicon. Because of its framework-independence and compactness, as explained in (Kinyon, 2003), we believe the MG to be a promising candidate for grammar extraction, because one can generate, from an extracted MG, grammars for different frameworks.

Acknowledgements We thank Olga Babko Malaya and Alexandrin Popescul for very helpful discussions on Russian, Aravind Joshi, Julia Hockenmaier, Anna Kupsc for helpful comments and discussions on earlier versions of this work.

References

A. Abeillé, M. Candito, and A. Kinyon. 1999. FTAG: current status and parsing scheme. In *Proc. Vextal-99*, Venice.

E. Bender, D. Flickinger, and S. Oepen. 2002. The Grammar Matrix: an open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proc. GEE-COLING*, Taipei.

M. Butt, S. Dipper, A. Frank, and T. Holloway-King. 1999. Writing large-scale parallel grammars for English, French, and German. In *Proc. LFG-99*.

M. Butt, H. Dyvik, T.H. King, H. Masuichi, and C. Rohrer. 2002. The parallel grammar project. In *proc. GEE-COLING*, Taipei.

M.H. Candito. 1996. A principle-based hierarchical representation of LTAGs. In *Proc. COLING-96*, Copenhagen.

M.H. Candito. 1999. *Représentation modulaire et paramétrable de grammaires électroniques lexicalisées*. Ph.D. thesis, Univ. Paris 7.

L. Clément and A. Kinyon. 2003a. Generating LFGs with a MetaGrammar. In *Proc. LFG-03*, Saratoga Springs.

L. Clément and A. Kinyon. 2003b. Generating parallel multilingual LFG-TAG grammars with a MetaGrammar. In *Proc. ACL-03*, Sapporo.

L. Clément and A. Kinyon. 2001. XLFG: an LFG parsing scheme for french. In *Proc LFG-01*, Hong-Kong.

M. Dalrymple, J. Lamping, F. Pereira, and V. Saraswat. 1995. Linear logic for meaning assembly. In *Proc. CLNLP*, Edinburgh.

D. Flickinger. 1987. *Lexical rules in the hierarchical lexicon*. Ph.D. thesis, Stanford.

A. Frank. 2000. Automatic F-Structure annotation of treebank trees. In *Proc. LFG-00*, Berkeley.

B. Gaiffe, B. Crabbé, and A. Roussanaly. 2002. A new metagrammar compiler. In *Proc. TAG+6*, Venice.

K. Gerdes. 2002. DTAG. attempt to generate a useful tag for german using a metagrammar. In *Proc. TAG+6*, Venice.

A. K. Joshi and K. Vijay-Shanker. 1989. Treatment of long distance dependencies in LFG and TAG: Functional uncertainty in LFG is a corollary in TAG. In *Proc. ACL-89*, Vancouver.

M. Kameyama. 1986. Characterising LFG in terms of TAG. In *Unpublished Manuscript*, Univ. of Pennsylvania.

R. Kaplan and J. Maxwell. 1996. LFG grammar writer's workbench. Technical Report version 3.1, Xerox corporation.

R Kaplan and A. Zaenen. 1989. Long distance dependencies, constituent structure and functional uncertainty. In *Alternatives conceptions of phrase-structure*, Univ. of Chicago press.

T. King. 1993. *Configuring topic and focus in Russian*. Ph.D. thesis, Standford Univ.

A. Kinyon and C. Prolo. 2002. A classification of grammar development strategies. In *Proc. GEE-COLING*, Taipei.

A. Kinyon and O. Rambow. 2003. Using the MetaGrammar to generate cross-language and cross-framework annotated test-suites. In *Proc. LINC-EACL*, Budapest.

A. Kinyon. 2000. Hypertags. In *Proc. COLING-00*, Saarebrucken.

A. Kinyon. 2003. *MetaGrammars for efficient development, extraction and generation of parallel grammars*. Ph.D. thesis, Proposal. Univ. of Pennsylvania.

J. Kuhn. 1998. Resource sensitivity in the Syntax-Semantic interface and the German Split NP construction. In *Proc. ESSLLI workshop: current topics in constraint-based theories of German syntax*, Saarbrucken.

B. Srinivas. 1997. *Complexity of lexical descriptions and its relevance for partial parsing*. Ph.D. thesis, Univ. of Pennsylvania.

F. Xia. 2001. *Automatic grammar generation from two perspectives*. Ph.D. thesis, Univ. of Pennsylvania.

Deep Grammar Development for Modern Greek

Valia Kordoni
Department of
Computational Linguistics
Saarland University
kordoni@coli.uni-sb.de

Julia Neu
Department of
Computational Linguistics
Saarland University
june@coli.uni-sb.de

Abstract

We present a large grammar fragment of Modern Greek. The grammar is written in the HPSG formalism and is being developed in a multilingual context with MRS semantics.

1 Introduction

In this paper we describe the development of a large grammar fragment of Modern Greek in a multilingual context. The grammar is couched in the theoretical framework of HPSG (Head-Driven Phrase Structure Grammar; (Pollard and Sag, 1994)) and benefits from an organization of semantics based on MRS (Minimal Recursion Semantics; (Copestake et al., 1999), (Copestake et al., 2001)).

MRS, a framework for computational semantics, in which the meaning of expressions is represented as a flat bag of Elementary Predications (EPs), combines naturally with typed feature structures, like the ones used in HPSG, and allows for structures which are underspecified for scopal information and can be compared across languages. HPSG itself is also suitable for multilingual grammar development, since the analyses in the extensive literature written in it can be shared across languages, but also parametrized accordingly, and its characteristic type hierarchy enables the writing of grammars that are easy to extend. Moreover, there are by now many useful open-source tools for writing, testing, and efficiently processing grammars written in HPSG and MRS: the LKB system for grammar development (Copestake, 2002), [incr tsdb()] for testing grammars and tracking changes (Oepen and Carroll, 2000), and PET, a very efficient HPSG parser for processing (Callmeier, 2000).

The tool we use for the development, i.e., the writing and the testing of the Modern Greek grammar is the Grammar Matrix (Bender et al., 2002), an open source tool ("starter-kit") designed for the rapid development of multilingual broad coverage grammars couched in HPSG and MRS and based on LKB.

As described thoroughly in (Bender et al., 2002), some of the very important and useful components of the Grammar Matrix include types defining the basic feature geometry necessary for starting off with the development of a grammar fragment in a given language couched in HPSG, types associated with MRS, a number of rules which reflect the general principles of HPSG (e.g., the Head Feature Principle, the Nonlocal Feature Principle, and so forth), constructional types, which correspond to the Immediate Dominance (ID) Schemata of HPSG (e.g., the head subject schema, the head complement schema, and so forth), and to more specific constructions, such as relative clauses (see (Sag, 1997)), as well as configuration and parameter files for the LKB grammar engineering environment (see (Copestake, 2002)).

It is important to underline here that the grammar writer does not have to make use of all the different components of the Grammar Matrix briefly mentioned above. For the grammar writer the

Grammar Matrix is the necessary complete and robust platform, which offers the appropriate flexibility and support needed right from the beginning of the implementation of a grammar fragment for a new language.

Indeed, from the experience we have gathered during the different stages of the development of the deep Modern Greek grammar it is clear that some of the components of the Grammar Matrix might only be important at a later stage of the grammar development. That is, the Grammar Matrix is obviously designed in order not only to provide the grammar writer with a smooth and quick start, but also to facilitate the long-term development of a large-scale HPSG grammar.

In the following we focus on some detailed examples from the deep Modern Greek grammar we have been developing since January 2003 using the Grammar Matrix, as part of the DELPHIN Collaboration (Deep Linguistic Processing with HPSG: An International Collaboration; for more see `http://www.delph-in.net/`), which currently involves research groups from DFKI in Saarbrücken, Saarland University, Stanford University, Tokyo University, the University of Sussex, Cambridge University, and the University of Trondheim, and whose main current research takes place in three areas:

- Area One: Robustness, Disambiguation and Specificity of HPSG Processing
- Area Two: The application of HPSG Processing to Information Extraction
- Area Three: Multilingual Grammar Engineering

aiming mainly at the further promotion of the central role that robust and deep processing of natural language in a multilingual context based on HPSG plays nowadays in human language technology.

Our aim here is not only to present some interesting aspects of the HPSG grammar of Modern Greek currently under development, but also to show the practical support we have drawn from the Grammar Matrix platform, which has enabled us to concentrate on the implementation of very demanding Modern Greek linguistic data right from the beginning of our project.

2 Modern Greek HPSG Syntax

The fundamental notion of a grammar written in HPSG is the sign, a complex feature structure which conveys phonological, syntactic, semantic, pragmatic and discourse information at the same time. The attribute-value matrix of a sign in the Modern Greek HPSG grammar is somewhat similar to a sign in the LinGO English Resource Grammar (ERG; (Flickinger, 2000)), as well as the HPSG Japanese Grammar (see (Siegel and Bender, 2002)), with information about the orthographical realization of the lexical sign in STEM, syntactic and semantic information in SYNSEM, nonlocal information in NONLOC, head information that percolates up the tree structure via HEAD and subcategorization information in VAL(ENCE) (whose values are SUBJ, COMPS, SPEC, and SPR).

Features, such as STEM, SYNSEM, NONLOC, HEAD, VAL(ENCE), INFLECTED (for inflection), or ARGS (for the different types of arguments), for instance, which are part of the sign geometry, as well as a large number of types which inherit from sign itself, are already defined in the Grammar Matrix.

For the implementation of the Modern Greek HPSG grammar we only needed to define further the parts of speech which are relevant to Modern Greek along with their specific head features. To these belong determiners, nouns, pronouns, affixes, prepositions and verbs, all of which inherit directly or indirectly from (the type) `sign` in the Grammar Matrix. Figure 1, for instance, shows the type for determiners implemented in the Modern Greek grammar.

```
det-lxm := notinflect-word-or-lexrule &
[ SYNSEM.LOCAL.CAT [ HEAD det,
                     VAL [ SUBJ < >,
                           SPR < >,
                           COMPS < >,
                           SPEC < > ]],
  SYNSEM.LOCAL [ KEYS.KEY #key & [ BV ref-ind ],
                 CONT [ LISZT <! #key !>,
                        INDEX ref-ind,
                        H-CONS <! !> ]]].
```

Figure 1: Type for determiners including semantic information

As far as the subcategorization patterns of verbal predicates in Modern Greek are concerned, it has turned out that for the best part these can be accounted for in the implementation of the Modern Greek grammar fragment by relying on the material that the Grammar Matrix already provides for. The valence list for verbs in the Grammar Matrix contains a SUBJ list and a COMPS list. These lists are also appropriate for all verbal classes in Modern Greek.

The grammar implementation is based on a system of types. 4 months from the beginning of the project, i.e., in May 2003, the grammar covers a good range of constructions in Modern Greek. Due to space limitations, though, only some of these phenomena can be described here.[1]

2.1 Verbal Predicates

As mentioned briefly above, the implementation of all the basic verbal predicate constructions of Modern Greek has been central to our project so far.

Types for intransitive verbs like *treho* (to run)

(1) To pedi trehi.
the child.N run.PRES.3S
"The child is running".

as well as for transitive verbs which take a direct object as the single element on their complement list, such as *fonazo* (to call)

(2) O Petros fonazei to koritsi.
the Peter.N call.PRES.3S the girl.A
"Peter calls the girl".

have been among the first ones defined in the Modern Greek grammar.

The grammar implementation has been since extended accordingly in order to treat also "more demanding" verbal classes in Modern Greek, such as valence alternating verbs, which support direct internal arguments (i.e., objects), as well as indirect prepositional arguments. Such alternation patterns in Modern Greek characterize among others the behaviour of verbal predicates which participate in the so-called Locative Alternation phenomena (cf., (Levin and Rappaport Hovav, 1991) for similar constructions in English):

[1]For a more detailed description of all the phenomena covered in the fragment of Modern Greek see (Neu and Kordoni, 2003).

(3) O georgos fortose to ahiro sto
the farmer.N load.PAST.3S the hay.A onto-the
karo.
wagon
"The farmer loaded the hay on the wagon".

(4) O georgos fortose to karo me ahiro.
the farmer.N load.PAST.3S the wagon.A with hay
"The farmer loaded the wagon with hay".

(5) O Petros adiase tin dexameni apo to
the Peter.N empty.PAST.3S the tank.A of the
nero.
water
"Peter emptied the tank of water".

(6) O Petros adiase to nero apo tin
the Peter.N empty.PAST.3S the water.A from the
dexameni.
tank
"Peter emptied the water from the tank".

For the implementation of such verbs, which draws on the theoretical HPSG analyses proposed in the literature (cf., (Kordoni, 2003)), we assume the following three verbal classes:

- Class 1: verbs which take either a location or a locatum as their direct object;
- Class 2: verbs which only support a location argument as their direct object;
- Class 3: verbs whose direct object must be a locatum argument.

Moreover, we assume a general type for transitive verbs which allows for verbs to support also a PP argument (see Figure 2).

```
trans-pp-basic := verb-lxm &
[SYNSEM.LOCAL.CAT
  [HEAD verb,
   VAL [SUBJ <[SYNSEM.LOCAL [CAT [HEAD noun-or-pronoun,
                                  VAL.SPR < >],
                             CONT.INDEX #arg1]]>,
        COMPS <[SYNSEM.LOCAL [CAT [HEAD noun &
                                   [QUALIA qualia]],
                              CONT.INDEX #arg3]],
               [SYNSEM.LOCAL [CAT [HEAD prep &
                                   [PTYPE ptype],
                                   VAL.SPR < >],
                              KEYS.KEY.ARG #e,
                              CONT.INDEX #arg-pp]]>],
   ARG-ST <[SYNSEM.LOCAL.CAT.HEAD affix]>],
 SYNSEM.LOCAL [KEYS.KEY #key & arg13-PP-rel &
                              [EVENT #e,
                               ARG1 #arg1 & ref-ind,
                               ARG3 #arg3 & ref-ind,
                               ARG-PP #arg-pp],
               CONT.LISZT <! #key !>]].
```

Figure 2: The `trans-pp-basic` type of the Modern Greek grammar

The type in Figure 2 indicates that the verb subcategorizes for a subject and supports two elements on the COMPS-list: a direct object NP, whose INDEX is ARG3, and a PP, whose INDEX value is shared with that of ARG-PP. This feature we have added as part of a new relation *arg13-PP-rel*, which we have allowed to inherit from the *arg13-rel* of the Grammar Matrix, and which we have introduced in order to output the correct semantics every time the trans-pp-basic applies; that is, in order to enable the NP and the PP to be linked every time to the right semantic arguments in the MRS semantic representantions of sentences headed by verbs of type trans-pp-basic.

Note that the direct object NP of such verbs bears a feature QUALIA, whose value *qualia* is underspecified in Figure 2. *qualia* can be further subtyped as *location*, *locatum* and *no-qual* (in the case of animate nouns and pronouns). In the type trans-pp-basic of Figure 2 QUALIA remains underspecified, allowing for flexibility in the linking of the NP to different semantic arguments according to the semantic requirements associated with the direct object NP of verbs in each of the three implementation classes above.

Finally, the head feature PTYPE we have introduced for prepositions is also underspecified in the type trans-pp-basic in Figure 2. All the prepositions in Modern Greek are subtypes of *ptype*. PTYPE gets accordingly specified depending on the semantic requirements associated with the PP argument of verbs in each of the three implementation classes above.

2.1.1 Verbs of Class 1

Due to space limitations, we only discuss here the verbs we have implemented under Class 1, as this has been described in Section (2.1) above. These are valence alternating, locative verbs in Modern Greek whose direct object may denote either a *location* or a *locatum* (see examples (3)-(6) in Section (2.1) above).

The type in our grammar which accounts for verbs like the ones in (3) and (4) inherits from *trans-pp-basic* in Figure 2 above, with a further specification only for the values of QUALIA[2] and PTYPE. Figure 3 captures the properties of the type appropriate for verbs of Class 1 which combine with the prepositions *"me"* (with) or *"sto"* (onto).[3]

```
trans-class1-me-or-sto-lxm := trans-pp-basic &
[ SYNSEM.LOCAL.CAT.VAL.COMPS
   < [ SYNSEM.LOCAL.CAT.HEAD.QUALIA qualia ],
     [ SYNSEM.LOCAL.CAT.HEAD.PTYPE me-or-sto ] > ].
```

Figure 3: Type in the Modern Greek grammar appropriate for verbs of Class 1 which combine with the prepositions *"me"* (with) or *"sto"* (onto)

For verbs like the ones in (5) and (6) in Section (2.1) above, which may only combine with a PP headed by the preposition *"apo"* (of/from), the appropriate type defined in the Modern Greek grammar fragment is similar to the one in Figure 3. The only difference in the verb type is the value of PTYPE.

```
trans-class1-apo-lxm := trans-pp-basic &
[ SYNSEM.LOCAL.CAT.VAL.COMPS
     < [ SYNSEM.LOCAL.CAT.HEAD.QUALIA qualia ],
       [ SYNSEM.LOCAL.CAT.HEAD.PTYPE apo ] > ].
```

Figure 4: Type in the Modern Greek grammar appropriate for verbs of Class 1 which combine with the preposition *"apo"* (of/from)

2.2 Cliticization Phenomena

Central to our project so far has also been the implementation of clitics and phenomena related to clitics in Modern Greek, such as clitic doubling phenomena (cf., (Kordoni, 2001)).

In general, Modern Greek distinguishes mainly between genitive and accusative clitics.

Clitics in Modern Greek share a significant number of properties with what have been argued to be *pronominal affixes* in some Romance languages, such as French, Italian, and Portuguese (cf., (Miller and Sag, 1997), (Monachesi, 1996), (Crysmann, 1997), respectively).

[2]In Figures 3 and 4 QUALIA is still underspecified as verbs of this class admit either a location or a locatum semantic argument as direct object.

[3]In this case we avoid overgeneration by imposing a further restriction on prepositions: prepositions get a further head feature TAKEQUAL whose value is of type *qualia* and is also shared with the QUALIA value of the NP they combine with. For example, the TAKEQUAL feature of the preposition *me* gets the value *locatum*, thus ensuring that *me* can combine only with nouns of type *locatum*.

That is, they cannot be topicalized, they cannot be substituted by full pronouns, they cannot be coordinated, and they cannot be modified. In addition, applying to Modern Greek "clitics" the criteria of (Zwicky and Pullum, 1983) for distinguishing affixes from postlexical clitics, we come to the conclusion that the Modern Greek entities in question are indeed *pronominal affixes* rather than *pronominal clitics*, which should not be treated exclusively as independent syntactic entities.

Modern Greek pronominal affixes alternate between preverbal and postverbal ones, or *prefixal* and *suffixal*, respectively. In Modern Greek this alternation depends on whether the verbal host of the pronominal affix is in the indicative/subjunctive, or the imperative form. That is, *suffixation* is triggered *only* in the cases when pronominal affixes co-occur with a verb in the imperative non-periphrastic form:

(7) Pesto.
say.IMP.2S-cl.A
"Say it."

(8) *To pes. (with the imperative meaning)
cl.A say.IMP.2S
"Say it."

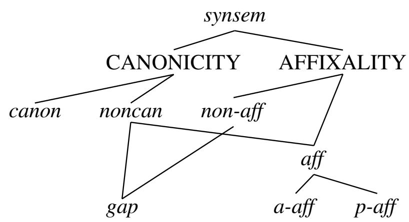
while in the cases when pronominal affixes co-occur with a verb in the imperative periphrastic or the indicative/subjunctive form, *prefixation* is triggered. The periphrastic form of the imperative in Modern Greek consists of the complementizer *na* immediately followed by the verb either in the Present Indicative or in the Present Subjunctive form:

(9) Na to dinis ligo ligo piso to danio
to cl.A give.IND.2S bit bit back the loan.A
stin trapeza.
to-the bank
"You should pay the loan back to the bank bit by bit."

(10) *Na dinis-to ligo ligo piso to danio
to give.IND.2S-cl.A bit bit back the loan.A
stin trapeza.
to-the bank
"You should pay the loan back to the bank bit by bit."

The implementation of Modern Greek pronominal affixes in the grammar fragment described here draws on mainstream and well-established theoretical HPSG proposals, according to which in HPSG words may come along with an argument structure (ARG-ST), which is an attribute that determines the combinatorial potential of a word, including specific subcategorization restrictions (cf., (Manning and Sag, 1999)). The members of ARG-ST may be of sort *canonical* or of sort *noncanonical* (i.e., *gaps* or *affixes*; see also (11)). As shown in (11), *canon(ical-synsem)*

> "is the type associated with all signs; *noncan(onical-synsem)* corresponds to an ARG-ST position that is not realized as a local syntactic dependent of the head. The latter subtype is in turn divided into the subtype *aff(ixal-synsem)* and *gap(-synsem)*. It is the presence of elements of type *aff* on a verb's ARG-ST list that triggers the morphological realization of the corresponding pronominal affixes. The type *non-aff* provides a cross-cutting classification, subsuming all types of *synsem* other than *aff*" (Miller and Sag, 1997).

(11)


In the Modern Greek grammar fragment pronominal affixes are defined as follows:

```
affix-lxm := notinflect-word-or-lexrule &
[ SYNSEM.LOCAL.CAT [ HEAD affix &
                          [ CASE case ],
                     VAL [ SUBJ < >,
                           COMPS < >,
                           SPEC < >,
                           SPR < > ]],
  SYNSEM.LOCAL [ KEYS.KEY #key & [ INST #inst ],
                 CONT [ LISZT <! #key !>,
                        INDEX #inst,
                        H-CONS <! !> ]]].
```

Figure 5: Type for affixes in the Modern Greek grammar

For the implementation of pronominal affixes in clitic doubling constructions in Modern Greek:

(12) O Petros to fonazei to koritsi.
the Peter.N cl.A call.3S the girl.A
"Peter is calling the girl".

we have introduced a new rule in the grammar fragment, the *clitic doubling rule*.

The *clitic doubling rule*[4] inherits from the type `head-final` of the Grammar Matrix, and it enables a verb (the head) and an adjacent affix appearing on the left of the verb to combine (see Figure 6).

The verb comes along with an ARG-ST list, whose lone element is token-identical to the synsem of the affix. After combining with the affix, the ARG-ST list of the verb is empty. The COMPS list of the verb remains unchanged; it just gets fully copied to the projection V' (see Figure 6), since at this stage the verb still needs to combine with a direct object (i.e., the "clitic-doubled" NP *to koritsi* in example (12)), and potentially with more (optional) arguments. Finally, the values of the verb's SUBJ, SPR and SPEC are also copied to the projection V' (see Figure 6).

In order to restrict the agreement possibilities of the affix, we restrict the features PERSON, NUMBER, GENDER and CASE of the affix accordingly, so that the affix be coindexed with the first element of the COMPS list of the verb (i.e., coindexed with the direct object NP).

The information carried by the affix in the semantic representation of Modern Greek clitic doubling constructions in the grammar fragment is related to the clitic doubling rule. The value of the INST of the affix is token-identical to the value of the corresponding AFFIX feature of the verb.[5]

Thus, in the semantic representation of clitic doubling constructions in Modern Greek like the one in example (12), the verb is represented bearing a "slot" for an affix whose value is shared with the INST value of the affix itself; that is, in the semantic representation the affix appears as a direct dependent of the verb (for more see next Section).

3 MRS Semantics for Modern Greek

The large grammar fragment of Modern Greek, some important aspects of which we describe here, benefits from an organization of semantics based

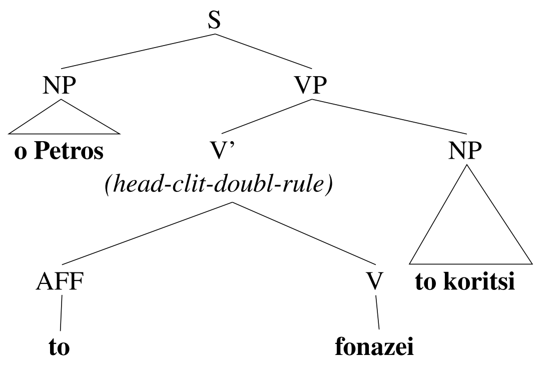


Figure 6: Tree representation for the sentence "*O Petros to fonazei to koritsi*"

on MRS (Minimal Recursion Semantics; (Copestake et al., 1999), (Copestake et al., 2001)). Types associated with MRS are already included in the Grammar Matrix tool we use for the development of the Modern Greek grammar.

As already mentioned in Section (1), MRS, a framework for computational semantics, in which the meaning of expressions is represented as a flat bag of Elementary Predications (EPs) encoded as values of a LISZT attribute, combines naturally with typed feature structures, like the ones used in HPSG, and allows for structures which are underspecified for scopal information and easily comparable across languages, thus being very appropriate for grammars, like the one we describe here, which are developed in a multilingual context, as part of an international initiative for multilingual deep linguistic processing based on HPSG (cf., Delphin Collaboration in Section (1)). Due to space limitations, here we discuss only the semantics related to the syntactic phenomena we have presented in Sections (2.1.1) and (2.2).

3.1 Valence Alternations in Modern Greek

The primary issue in the analysis of Modern Greek locative alternating verbal predicates like the ones in sentences (3) and (4) of Section (2.1) is to get the semantics right: that is, the two valence patterns should have the NP and the PP linked each time to the right semantic arguments.

(13) and (14) show how this issue is handled in the semantics of the Modern Greek grammar fragment.

[4]Due to space limitations we do not show the *clitic doubling rule* here.

[5]The AFFIX feature of the verb has been introduced as part of a new relation *arg13-aff-rel*, which inherits from the relation *arg13-rel* of the Grammar Matrix.

(13) Semantic representation in (indexed) MRS of the sentence in (3):

<h1,e2:TENSE:IND:NO_ASPECT,
{ **o** (x3:PNG, h4, h5),
georgos (x3, h6),
fortono (e2,v10,e9:NO_TENSE:ASPECT:MOOD,x3,x8:PNG,h7),
to (x8, h11, h12),
ahiro (x8, h13),
sto (e9, e2, x15:PNG, h14),
contracted_det (x15, h16, h17),
karo (x15, h18),
prpstn (h19, h1) },
{ h4 QEQ h6,
h11 QEQ h13,
h16 QEQ h18,
h19 QEQ h7 } >

(14) Semantic representation in (indexed) MRS of the sentence in (4):

<h1,e2:TENSE:IND:NO_ASPECT,
{ **o** (x3:PNG, h4, h5),
georgos (x3, h6),
fortono (e2,v10,e9:NO_TENSE:ASPECT:MOOD,x3,x8:PNG,h7),
to (x8, h11, h12),
karo (x8, h13),
me (e9, e2, x15:PNG, h14),
ahiro (x15, h16),
udef (x15, h17, h18),
prpstn (h19, h1) },
{ h4 QEQ h6,
h11 QEQ h13,
h17 QEQ h16,
h19 QEQ h7 } >

In the MRS semantic representations of the sentences in (3) and (4) of Section (2.1) that we have shown in (13) and (14) above, *h1* is the top handle of the sentence, which appears again as handle of **prpstn** (proposition; this conveys that the sentence is a proposition, rather than a question, for instance). Each noun gets an instantiation variable x_n (see *x3*, *x8*, and so forth in (13) and (14) above), which is bound to the respective determiners of the nouns. In case there is a contracted determiner (like, for instance, in (13) above), or no determiner at all (like, for instance, in (14) above), x gets bound to an appropriate relation (**contracted_det** or **udef**, respectively). Each entry in the MRS semantic representations in (13) and (14) above gets a handle which may be part of a QEQ-constraint of the H-CONS list.[6] The verbal heads in both (13) and (14) above come along with three arguments: ARG1, ARG3 and ARG-PP. These arguments are denoted by variables which are shared with those of the corresponding noun entries in the LISZT. It should be noted that the ARG-PP is denoted by an event variable (*e9*) which is shared with the corresponding variable of the preposition. The verbs in both (13) and (14) also allow for affixes, which, in this case, are indicated by a free variable (*v10*), because there are no affixes in the sentences in (3) and (4) of Section (2.1). The H-CONS list includes the QEQ-constraints which hold between the nouns and their determiners, as well as between the SOA of the **prpstn** (*h19*) and the handle of the verbs (*h7*).

[6]For more on QEQ-constraints and H-CONS lists see (Copestake et al., 1999) and (Copestake et al., 2001).

3.2 Clitic Doubling in Modern Greek

The analysis of Modern Greek clitic doubling constructions (see example (12) and Figure 6 in Section (2.2) above) has led us to innovations in the correspondence between semantic and syntactic structures in comparison to the Grammar Matrix.

Consider the semantic representation of the sentence in (12) of Section (2.2) presented in (15).

(15) Semantic representation in (indexed) MRS of the sentence in (12):

<h1,e2:TENSE:IND:NO_ASPECT,
{ **o** (x3:PNG, h4, h5),
petros (x3, h6),
to (v7, h8),
fonazo (e2, v7, x3, x10:PNG, h9),
to (x10, h11, h12),
koritsi (x10, h13),
prpstn (h14, h1) },
{ h4 QEQ h6,
h11 QEQ h13,
h14 QEQ h9 } >

Note that the affix entry bears a variable (*v7*), which is identical to the variable denoting the affix dependent of the verb. The handle of the affix, though, does not contribute to the QEQ-constraints. The affix has been combined with the verb by means of the clitic doubling rule (see Figure 6), and not by means of the head complement rule, a fact which is mirrored in the way this combination of the affix and the verb is denoted in the semantic representation of the sentence in (12) shown in (15): the affix is not a “real argument” of the verb; it is part of the morphosyntactic features of the verb. An alternative to the syntactic and semantic treatment of clitic doubling phenomena in Modern Greek that we propose here would have been to allow for the affix to combine with the verb in the definition of the type for transitive verbs (i.e., in `trans-lxm` in *my_language.tdl*). Such a treatment would not have gained us a more elegant solution, though.

4 Conclusion and Outlook

In this paper we have presented some important aspects of a large grammar fragment of Modern Greek we develop using the Grammar Matrix (Bender et al., 2002).

Our aim in this paper has been twofold: to present some interesting aspects of the HSPG grammar of Modern Greek, as well as to indicate which aspects of the Grammar Matrix as broad coverage grammar development platform are helpful and which have needed further development in the case of Modern Greek.

In future work, the next immediate steps in the development of the deep Modern Greek grammar include the implementation of word order phenomena, as well as the connection of the grammar system to a morphological analysis system and the incorporation of use of default entries for words unknown to the Modern Greek HPSG lexicon.

The Modern Greek HPSG grammar is accompanied by a detailed documentation (Neu and Kordoni, 2003), which is revised by the grammar writers every time new phenomena are incorporated in the grammar and which aims at ensuring reusability of the grammar itself.

References

Emily Bender, Dan Flickinger, and Stephan Oepen. 2002. The Grammar Matrix: An Open-Source Starter-Kit for the Rapid Development of Cross-Linguistically Consistent Broad-Coverage Precision Grammars. In John Carroll, Nelleke Oostdijk, and Richard Sutcliffe, editors, *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics. Taipei, Taiwan*, pages 8–14.

Ulrich Callmeier. 2000. Pet – a platform for experimentation with efficient HPSG processing techniques. *Journal of Natural Language Engineering 6(1): Special Issue on Efficient Processing with HPSG: Methods, Systems, Evaluation*, pages 99–108.

Ann Copestake, Dan Flickinger, Ivan A. Sag, and Carl J. Pollard. 1999. Minimal Recursion Semantics: An Introduction. Ms., Stanford University.

Ann Copestake, Alex Lascarides, and Dan Flickinger. 2001. An Algebra for Semantic Construction in Constraint-based Grammars. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL 2001), Toulouse, France*.

Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Lecture Notes, Number 110. Standord: CSLI Publications.

Berthold Crysmann. 1997. Parallel Morpho-Syntactic Constraints in European Portuguese Cliticization. Paper presented at the *Fourth International Conference on Head-driven Phrase Structure Grammar (HPSG)*, July 18-20, Cornell University, Ithaca, NY.

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Journal of Natural Language Engineering 6(1): Special Issue on Efficient Processing with HPSG: Methods, Systems, Evaluation*, pages 15–28.

Valia Kordoni. 2001. *Psych Verb Constructions in Modern Greek: a semantic analysis in the Hierarchical Lexicon*. Ph.D. thesis, University of Essex, Colchester, UK.

Valia Kordoni. 2003. Valence Alternations in Modern Greek: an MRS Analysis. In Jong-Bok Kim and Stephen Wechsler, editors, *The Proceedings of the 9th International Conference on Head-Driven Phrase Structure Grammar, Kyung Hee University, Seoul*, pages 129–146. Stanford: CSLI Publications, ISSN 1535-1793.

Beth Levin and Malka Rappaport Hovav. 1991. Wiping the Slate Clean: A Lexical Semantic Exploration. In Beth Levin and Steven Pinker, editors, *Lexical and Conceptual Semantics*, pages 123–152. Blackwell, Cambridge MA and Oxford UK.

Christopher D. Manning and Ivan A. Sag. 1999. Dissociations between Argument Structure and Grammatical Relations. In Gert Webelhuth, Jean-Pierre Koenig, and Andreas Kathol, editors, *Lexical And Constructional Aspects of Linguistic Explanation*, pages 63–78. CSLI Publications, Stanford, Calif.

Philippe H. Miller and Ivan A. Sag. 1997. French clitic movement without clitics or movement. *Natural Language and Linguistic Theory*, 15:573–639.

Paola Monachesi. 1996. *The syntax of Italian clitics*. Ph.D. thesis, University of Tilburg, Tilburg.

Julia Neu and Valia Kordoni. 2003. Implementation of a Modern Greek Grammar Fragment in HPSG. Technical report, Department of Computational Linguistics, University of Saarland.

Stephan Oepen and John Carroll. 2000. Performance Profiling for Parser Engineering. *Journal of Natural Language Engineering 6(1): Special Issue on Efficient Processing with HPSG: Methods, Systems, Evaluation*, pages 81–97.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago.

Ivan A. Sag. 1997. English Relative Clause Constructions. *Journal of Linguistics*, 33.2:431–484.

Melanie Siegel and Emily M. Bender. 2002. Efficient Deep Processing of Japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization, Coling 2002 Post-Conference Workshop*. Taipei, Taiwan.

Arnold M. Zwicky and Geoffrey K. Pullum. 1983. Cliticization vs. inflection: English n't'. *Language*, 59:502–513.

3-Phase Grammar Learning

Geert-Jan M. Kruijff
Computational Linguistics, Saarland University
Saarbrücken, Germany
⟨gj@coli.uni-sb.de⟩

Abstract

In this paper we discuss an approach to learning grammars from treebanks that aims at being independent of a specific grammar framework, and capable of learning grammars for any type of language. We focus on learning word order models, presenting the approach and experiments on treebanks for English, Dutch and German. We illustrate how the approach can be used to construct a computational typology of word order, and can assist in building multilingual grammars.

Start with the things that you know and the things that are unknown will be revealed to you
– Rembrandt, 1609-1669

1 Introduction

More and more treebanks are becoming available, often for languages that are structurally quite different. This appears to necessitate that if we want to be able to learn grammars for these languages, we need an approach that makes as few assumptions as possible about the structures it needs to learn. Furthermore, if we are interested in grammars not as an *end* by themselves but as a *means* to study language, then it would be desirable if these grammars are linguistically plausible and cross-linguistically comparable.

In this paper we describe an approach to learning word order models set in *3-Phase Grammar Learning*, a framework to grammar learning that is *language independent*, *theory neutral*, and *typologically inspired*. Language independence and theory neutrality are practical points inspired by (Skut et al., 1997). They enable us to learn grammars for arbitrary types of languages, and specify the results into any lexicalized grammar framework. The typological orientation of our approach makes cross-linguistic comparison possible, and lines up with recent research in various contemporary grammar frameworks like LFG, categorial grammar, and HPSG.[1] A practical consequence of the symbiosis of all the above aspects of our approach is that becomes in principle possible to compare grammars across languages, and integrate into *multilingual* grammars (be they completely learned or partially hand-crafted).

3-Phase Grammar Learning (or 3PGL for short) is characterized by the three consecutive phases in which grammars are learned: Extraction, generalization, and specification. In the first phase, we extract the tree structures that we generalize in the second phase to word order models and a lexical inheritance hierarchy. The lexical inheritance hierarchy describes types of subcategorization patterns, which are linked to the word order models in two ways: One, we link a subcategorization pattern to a *canonical order* in which the head and its arguments are realized, and two, we link a subcategorization pattern to a word order model that describes the word order restrictions and variability that hold for the (sub)tree in which the subcategorization pattern is used. We thus explicitly separate word order from subcategorization. This helps us avoid the problems that (Baldridge, 2002) observes for (Villavicencio, 2002).

[1] Under the typological view, we agree that languages may disagree on their structures. The idea is not to straightjacket languages into a single set of universal principles, but to discover principles that show how languages are similar or different.

Up to generalization, the learning procedures are language-independent and theory-neutral. The resulting grammar models can in principle be translated during the specification phase into a grammar for any (non-transformational, lexicalized) grammar framework. Its language-independence and theory-neutrality places 3PGL apart from for example (Xia et al., 2000). Xia et al explicitly require their training data to be in Penn TreeBank bracketed format, a format that is neither theory-independent not language-independent: E.g. the Prague Dependency Treebank (Hajič, 1998) cannot be completely translated into this format, primarily due to the high degree of word order freedom found in Czech. Separating out specification, and keeping subcategorization and word order apart (though linked), makes 3PGL close to MetaGrammar-based approaches (Candito, 1996) (particularly, Type D grammars in the classification of (Kinyon and Prolo, 2002)).

To illustrate 3PGL, we focus in this paper on the component for learning word order models. Word order is an important aspect in which languages may differ. We describe the algorithms used in the extraction and generalization phases for learning word order models, and then discuss how these models can be specified as *generalized field models*. Generalized field models are a generalization of topological field models, an approach to describing word order constraints that has been used in HPSG, Topological Dependency Grammar (Duchier and Debusmann, 2001), and categorial grammar (Kruijff, 2001).

Overview An overview of the paper is as follows. In §2 we discuss are approach to learning word order models. In §3 we describe how to relate these models to a lexical inheritance hierarchy, modeling subcategorization. Issues in cross-linguistic comparison and integration are discussed in §4. The paper closes with conclusions and issues for further research.

2 Learning word order models

In this section we discuss how we learn word order models in 3PGL. The extraction phase is relatively straightforward. Following approaches to constructing treebank grammars like (Charniak, 1996) or (Collins, 1999) we extract all the "productions" from a treebank. As a result we obtain sets of depth-1 trees, i.e. sequences of a head and all its modifications/immediate dependents, one set for each type of tree[2] In other words, from the viewpoint of word order modeling we thus assume that distinct tree types lead to distinct word order models.(See §2.2 for more discussion.)

Over these sets of sequences we then create models that generalize the word order patterns observed in the sequences. This sets our approach apart from more straightforward approaches like (Charniak, 1996) in which all the unique sequences simply constitute the productions in the grammar. Using the sequences without further generalization works reasonably well for English (Charniak, 2000), arguably because English shows relatively little word order variation – hence most if not all possible variations are likely to be observed in the training data. This leads both to a high coverage and high labeled f-measures – scores that are not equally met when applying the same approach to languages with a freer word order, like German (Dubey and Keller, 2003; Fissaha et al., 2003).[3]

We use *profile Hidden Markov Models* to learn generalizations over the training sequences. Profile HMMs are a specialized case of Hidden Markov Models. Their defining characteristic is a more refined ontology of internal states, which results in a model to represent not just a distribution over sequences, but also a characterization of what these sequences have in common (the *profile*, modeling consensus across sequences) and how they may differ. The important point is that this makes it possible for our approach to learn a fine-grained model of word order restrictions and variations. In this we differ for example from (Collins, 1997; Collins, 1999) in that the latter basically only uses a simple Markov Model to derive new

[2] We define *type* of a tree to be the type of nonterminal that governs the wordgroup (=phrase), or (in a pure dependency-based approach) the type of head that governs the word group.

[3] Currently, we cannot make any qualified statements. We can only hypothesize at the moment that, in the absence of further generalization, and using fixed quantities of training and test sequences to be able to compare results across languages, it is likely that coverage and labeled f-measures will drop further, the freer the word of a language is.

productions as generalizations. These new productions cannot be guaranteed to share common structures (constraints) with the original sequences they were learned from. As a result, such models are more likely to overgenerate.

In §2.1 below we describe the intuitions behind profile Hidden Markov Models. §2.2 discusses how we can evaluate the word order models we learn, and presents an evaluation of assigning models to sequences. In §2.3 we describe an algorithm for specifying profile Hidden Markov Models as generalized field models.

2.1 Profile Hidden Markov Models

Profile Hidden Markov Models (profile HMMs) are a specific version of Hidden Markov Models (HMMs) (Durbin et al., 1998). Firstly, they are *anchored*, having unique start and end states (Ristad, 1998). Secondly, they have a more fine-grained state ontology. The architecture of a profile HMM is built from three types of states: Match states, insert states, and delete states. The match states make up the profile, each match state having as emission symbols the structures that are found in the position that the match state models. Insert and delete states enable us to deal with sequences that have a different length than the profile (i.e. number of match states). An insert state can insert a structure (emission symbol) inbetween the structures of two match states, creating a longer sequence, whereas a delete state enables us to skip a match state, resulting in a shorter sequence. Figure 1 illustrates the architecture; squares are match states, diamonds insert states, and circles delete states. The left-most insert state, which can insert material before the first match state, is called the *free insertion state*.

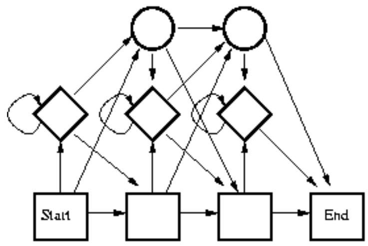


Figure 1: Profile HMM architecture

The result of distinguishing match, insert and delete states is that a profile HMM not only learns a a probability distribution over a possibly infinite number of sequences, but also what these sequences have structurally in common. Structural similarity is represented in the model as a positionwise consensus model across these sequences – the *profile*, constituted by the match states. The profile gives the distribution of structures shared across the training sequences, with one distribution per position (modeled by a match state) in an (optimal) alignment of sequences.

If we train a profile HMM over dependency trees of depth-1 (i.e. sequences) for a given tree types, then the model represents where different constituent types can occur. Modulo a threshold on emission probability, the distribution of a constituent type shows whether that type is restricted to occur in a single position (i.e. only in that state it has an emission probability above the threshold), or in multiple positions (i.e. in several states it has an emission probability above the threshold). Important to observe is that we can have restrictions for constituent type placement occurring for match states. In this case, such a restriction models a *general* word order restriction holding for all trees of the given tree type.

To illustrate the shape that word order restrictions and variations can take, consider Table 1 below. Table 1 provides a simplified fragment of a profile learned over NEGRA S-trees (matrix clauses). The columns are the match states, with their entropy given at the bottom lines. In each column, we give the emission probabilities for Subjects (SB), Accusative Objects (OA), Datives (DA), Clausal Objects (OC), PP Modifiers (MO), Adverbial Modifiers (MO[A]), and the verbal head (HD). The profile clearly illustrates verbsecondness as a restriction, the preference of the subject to occur close to the verbal head, and the tendency for heavier constituents like OCs to occur at the end of a clause.

Appendix A provides the basic definitions for profile HMMs, repeated from (Durbin et al., 1998).[4]

[4]Profile HMMs originated in bioinformatics, but the available implementations the set of emission symbols is restricted to 24 or 25 letters. Our implementation, consisting of a Java

0	1	2	3	4
SB: 0.35 OA: 0.04 DA: 0.01 OC: 0.05 MO: 0.22 MO[A]: 0.07	HD: 0.99	SB: 0.47 OA: 0.17 DA: 0.04 OC: 0.02 MO: 0.07 MO[A]: 0.10	SB: 0.13 OA: 0.13 DA: 0.03 OC: 0.03 MO: 0.20 MO[A]: 0.14	SB: 0.04 OA: 0.12 DA: 0.00 OC: 0.13 MO: 0.21 MO[A]: 0.01
2.32	0.01	1.92	2.69	2.46

Table 1: Simplified fragment of a profile for NEGRA S-trees

2.2 Evaluation

Before we even start thinking about evaluating the word order models learned by profile HMMs, we have to ask ourselves, What is it that we want to *know*? What should the evaluation *reveal*? Quantitatively, an important aspect of these models is whether they are *distinctive* enough: When presented with a sequence, can we assign the appropriate model to it, describing what word order constraints are to hold for that sequence to be well-formed? From a more qualitative viewpoint, it would of course be useful to know what kind of *coverage* the models have – what word order phenomena they do (or do not) model, and to what depth. And, somewhere in between qualitative and quantitative evaluations: How does the performance of a model degrade when we present it with sequences that violate one or more word order restrictions?[5]

To be able to perform a qualitative evaluation, however, we need a test suite. Unfortunately, current test suites do not appear to have a broad enough coverage of word order phenomena, nor do they usually model more than one or two languages. In §5 we present some ideas about how profile HMMs, and the word order constraints we can derive from them, could be combined with a MetaGrammmar-based approach (Kinyon and Rambow, 2003) to minimize human intervention in creating test suites.

The evaluation we present here is therefore purely quantitative. The goal of the evaluation is to investigate the distinctness of models, or (the other side of the coin), whether we can recognize the right tree type given the word order in a sequence. In the evaluation, we used three treebanks: The WSJ section of the Penn Treebank (Marcus et al., 1993), the Corpus Spoken Dutch (Oostdijk, 2000), and the NEGRA treebank for German (Skut et al., 1997). We translated the WSJ into the NEGRA Export Format used by the CSD and NEGRA, resulting in traceless, possibly non-projective dependency trees.[6] For each treebank, we extracted a collection of sets of sequences for those tree types modeling verbal constructions. Over each collection, we performed a ten-fold cross evaluation, distinguishing training and test parts. For each fold, the profile HMMs for a treebank were trained on the training data, and then tested on the test data. During testing, we assigned that model M to a sequence from the test data for which we had the highest Viterbi score, i.e. $max\ probability(sequence|Model)$.

Table 2 below shows for each treebank the scores averaged over the performance of each of the models (*average*), and that performance weighted by the number of sequences per test set (*weighted average*).

Treebank	Average	Weighted average
WSJ	0.891	0.800
CSD	0.716	0.760
NEGRA	0.997	0.997

Table 2: Overview model assignment: average over model performances, average model performance weighted by # sequences in test data sets

The results show that the profile HMMs learned on tree types for verbal constructions, when "competing" amongs one another for the highest score, do relatively well. A more detailed picture is presented in Tables 3–5 below. They show, per treebank, how often a model learned over the training data for tree type T, $M_T f$, gets assigned to sequences in the test data of T. There we see that the lower score for the CSD can be explained by the very low performance on SV1 tree types (verb-initial main clauses): It is almost always outperformed by SMAIN, i.e. SV1 is not distinct enough standard main clauses modeled by SMAIN.

API and a GUI workbench for creating training data, and learning, evaluating, and inspecting profile HMMs, allows for arbitrarily large, finite sets of emission symbols.

[5]In other words, how do our models behave in terms of *graded grammaticality* in comparison to e.g. (Keller, 2000)?

[6]We do realize that there is a difference in register between the WSJ and NEGRA on the one hand (newspaper texts) and the CSD on the other (spontaneous dialogues).

WSJ	S	SBAR	SBARQ	SQ	VP	VPn	VPto
Average	0.805	0.868	0.912	0.960	0.738	0.962	0.993
# test sequences	8771	2474	26	37	13407	1846	1606

Table 3: Overview model assignment in WSJ: average over model performances per tree type

CSD	INF	SMAIN	SSUB	SV1	SVAN	TI
Average	0.6732	0.998	0.633	0.01	0.983	0.994
#test sequences	537	1589	496	407	36	111

Table 4: Overview model assignment in CSD: average over model performances per tree type

2.3 From PHMMs to Generalized Field Models

If we want to specify the profile HMM word order models in terms of a particular grammar framework then a convenient stepping stone is to translate the models into *generalized field models* (GFMs). GFMs are based on the topological field model (TFM) that was initially developed for German, but can in principle be applied to any type of language (cf. e.g. (Penn, 1999; Kruijff, 2001)). The TFM is a theory-neutral model of clausal structure that provides a linguistically motivated, flat macro structure in terms of a total ordering over a small set of "fields" and "brackets" (Höhle, 1983). For each clause level, fields and brackets place constraints on cardinality (e.g. only one element in the Vorfeld) and eligible types (e.g. only finite verbs in the Left Bracket). The examples below illustrate a simple topological analysis of a German sentence, (using the traditional fields: VF=*Vorfeld*, LB=*Left Bracket*, MF=*Mittelfeld*, RB=*Right Bracket*).

(1) Kathy hat Elijah den Witz erzählt
Kathy has Elijah the joke told
"Kathy has told Elijah the joke."

(2) $\{$Kathy$\}_{vf}$ $\{$hat$\}_{lb}$ $\{$Elijah den Witz$\}_{mf}$ $\{$erzählt$\}_{rb}$

Topological fields have since been used in various frameworks (e.g. HPSG, Topological Dependency Grammar, CG) to give word order accounts. GFMs generalize the TFM as follows. We let any head have a GFM, not just clause levels, and we do not have a predefined set fields for a head – what number of fields we need, and what their characteristics are, is all learned from the data. Otherwise, GFMs are similar to the TFM. A *field* is a (possibly ordered) collection of constituent types; a GFM provides a string position-independent total ordering over fields. Restrictions and variations are modelled in terms of what fields a type of constituent can appear in, and what cardinality the individual fields have.

NEGRA	S	VP	VZ
Average	0.999	0.991	0.999
#test sequences	2897	1297	169

Table 5: Overview model assignment in NEGRA: average over model performances per tree type

It is relatively easy to derive a generalized field model from the match states only. By omitting the insert states, one makes the tacit assumption though that the material emitted by the match states is always *(string) adjacent*. In the approach we sketch below a generalized field model from the entire profile HMM; due to space restrictions we do not describe the formal algorithm.

There are essentially two issues we need to solve: When do we introduce new fields, and how do we determine cardinality?[7] To establish new fields, we first of all look at whether a particular insert or match state has *rigid* constituents, i.e. constituents that only in that state have an emission probability above a given emission threshold. If the match and insert states in a particular column have rigid constituents, then these constituents form a new field f_{rig}. If the set of variable (i.e. non-rigid) constituents of the insert state in the column, i_{var} is a subset of the set of variable constituents of the match state in that column, m_{var}, then we introduce a new field $f_{var} = m_{var}$; else, we add m_{var}, i_{var} separately, in that order. The field f_{rig} is added before the variable constituent field(s) if we are not in the last column of the profile HMM; else f_{rig} is added as the last field. The ordering over all the fields is established by the order in which they have been created.

To determine the cardinality of the respective fields, we first of all introduce two thresholds: A threshold on the reflexive transition from an insert state to itself, Θ_i, and a threshold Θ_d on the transition from the match state in column i to the delete

[7] In the discussion below we omit the free insertion state. Basically, this state can introduce fields for rigid and variable constituents; both will have a lower bound on their cardinality of 0.

state in column $i+1$. If the transition to the delete state in column $i+1$ is greater than Θ_d, then we set the lower bound on the cardinality of variable constituent fields introduced by states in column $i+1$ to 0; else it is 1. If the reflexive transition of the insert state in column i is greater than Θ_i, the upper bound on the cardinality of the field(s) that the variable constituents of the insert state end up in is set to "+"; else it is 2. The fields of rigid constituents introduced by insert/match states have a lower and upper bound of 1 on their cardinality.

3 Linking subcategorization and word order

As we already pointed out above, we distinguish subcategorization, represented by a lexical inheritance hierarchy, from word order models. In this division of labor, subcategorization defines what (and how) constituents can be put together to yield larger structures, whereas the word order models define how these structures should be linearized. (I.e. subcategorization provides us with the *resources* that we need to order.)

We construct a lexical inheritance hierarchy for a given tree type as a Galois lattice, similar to e.g. (Petersen, 2001; Sporleder, 2002). Each node in the hierarchy is defined for a *relation* and a set of one or more phrasal/part-of-speech types one out of which can be used to realize that relation. Each hierarchy has a *root*, defining the head element of the word group. The root node r can be modified by any choice of sister nodes s_i (types of dependents) that it immediately dominates. Furthermore, a node n immediately dominated by a node m can only be used to modify the root if also m is used to modify the root.

The root node of a lexical inheritance hierarchy for a tree type T is related to the profile HMM (or the derived GFM) for T. Furthermore, with each (non-root) node in the hierarchy, we node its *canonical ordering* relative to the head. Using a profile HMM, we determine the canonical ordering of a constituent (dependency relation plus specific phrasal/PoS-type) as follows. We start by normalizing all its emission probabilities across all the states it can occur in. The state for which we obtain the highest normalized emission probability is taken to be its canonical position. We can then derive canonical ordering of the constituent with respect to the head (i.e. before or after) by comparing the position of the canonical state of the dependent with respect to the canonical position of the head in the given model.

As a result, when constructing a subcategorization frame from a head and its dependents, we obtain at the same time a set of constraints that define canonical ordering of the dependents. We can then combine these canonical orderings with the profile HMM (or the GFM) for the tree type to obtain the (restricted) canonical field model, i.e. the canonical linearization of this subcategorization frame.[8]

Finally, we relate words to the lexical inheritance hierarchy by noting for each word the head-node and the minimal set of dependent-nodes it requires (i.e. its minimal frame, (Sarkar and Zeman, 2000)). If a dependent-nodes indicates that its relation can be realized through different types of phrases, we can indicate a preference order over the types. It is in this way we can model observed alternation preferences, and lexicalize canonical realizations (cf. also (Kurz et al., 2000)).

4 Cross-linguistic comparison and integration

4.1 Computational typology of word order

Typology is the study of universals in natural language: Starting from the viewpoint that languages in principle differ, we try to find out what characteristics groups of languages *share* and whether these shared characteristics can be implied by the presence of other, more fundamental characteristics of the inventory of a language.

Now that more and more electronic resources are available for an ever-widening range of languages, we can try to re-examine the typologies we have in a more *data-driven* way. Here, we would like to coin the term *computational typol-*

[8]Our approach thus differs from (Villavicencio, 2002) as follows. Villavicencio defines a lexical inheritance hierarchy in which categories in a categorial grammar are always extended monotonically, at the right-hand side (using a Steedman-type result-first notation). To obtain canonical ordering, Villavicencio introduces a principle of weak commutativity, with which arguments in a category thus formed can be re-ordered. As (Baldridge, 2002) shows, this approach does not always apply to languages other than English, and even for English the approach overgenerates.

ogy for that kind of effort.

One interesting application of the profile HMMs we have induced is that we can try to construct a computational typology of word order, starting with the classification (and comparison) on a scale describing the degree of word order freedom of a language. Traditionally, typology offers universals for canonical word order (after Greenberg, Hawkins). There are only a few proposals for universals describing degree of word order freedom (after Steele, Prague School; cf. (Kruijff, 2001)). These proposals invariably use a discrete scale: Either a binary division into "rigid" and "free" (Prague School), or a ternary division into "rigid", "mixed" and "free" (Steele, Kruijff). The problem with discrete scales is that languages in the same class are difficult to compare. For example, both Dutch and German are languages with a "mixed" degree of word order freedom (variable complement order, fixed positioning of the verb) yet native speakers do tend to consider Dutch less free than German.

Using profile HMMs modeling word order restrictions and variability, we can obtain a more fine-grained perspective by constructing a *continuous* scale to describe degree of word order freedom. The idea comes down to the following. Formal descriptions of word order usally fall between two extrremes. At one end, accounts start with a model of a completely rigid, canonical word order and then allow for rules to introduce *variations* on that order, e.g. categorial accounts like (Kruijff, 2001). On the other extreme, accounts start with a model that describes a completely free word order. Rules then act as constraints, imposing the necessary restrictions on word order; cf. e.g. (Duchier and Debusmann, 2001).

We use *both* perspectives here. We calculate the distance of a model to both a completely random model (the completely free case) and a canonical model (the completely rigid case), yielding a single score on a scale of 0 to 1. As a result, we obtain intervals of distance scores for each collection of models for a language. Table 6 shows the average score (*Average rigid/free*) in an interval, the minimum distance to the rigid model (i.e. how rigid a model can be) and the minimum distance to the free model (i.e. how free a model can be).

Treebank	Average rigid/free	Min. dist. rigid	Min. dist. free
WSJ (English)	0.278	0.054	0.505
CSD (Dutch)	0.361	0.211	0.583
NEGRA (German)	0.369	0.165	0.485

Table 6: Degrees of word order freedom

Despite the fact that there is a difference is register among the treebanks, we do see the traditional intuitions borne out. English is more rigid than either German or Dutch, with German allowing for more freedom than Dutch.

4.2 Towards learning multilingual grammars

The idea that we want to explore with 3PGL is whether we can in the end construct *multilingual* grammars, either by integrating manually crafted grammars with induced grammars, or just by combining induced grammars. We adopt thereby a strong notion of multilinguality: A (strongly) multilingual grammar is an architecture in which grammars for different languages share as much of their rules describing syntactic composition as possible.[9]

The strategy we thereby adopt is to use a typological orientation, first of all on word order, to establish what word order models can be combined.[10] The starting point thereby is the canonical field models, i.e. the generalized field models restricted to canonical ordering of the head and all its modifiers. Canonical models are the root nodes in a network of word order models, and are "extended" by nodes representing (shared) word order constraints (i.e. restrictions or variability).

The problem we need to tackle in this context is that of *treebank formats*: The kinds of trees that the treebank contains, i.e. pure phrase-structure including traces (Penn Treebank), pure dependency trees without phrases but with the possibility to have crossing edges (Prague dependency treebank), or (possibly non-projective) dependency trees with phrasal nodes (NEGRA, CSD); and, the kinds of phrases and Part-of-Speech tags that are being used. For example, the

[9] A weakly multilingual grammar would simply be a system in which we would have multiple grammars, without any interaction.

[10] This is akin to the multilingual networks of word order rules developed in (Kruijff, 2001).

finite verb of a clause is contained in the S phrase in NEGRA or the CSD, but is found in the VP in the WSJ.

5 Conclusions and future research

In this paper we discussed 3-Phase Grammar Learning, an approach to grammar learning that aims at being language-independent, theory-neutral, and that is inspired by language typology. We focused primarily on learning word order models, presenting a probabilistic approach to learn models of sets of trees for a given tree type, and describing an algorithm that can transform these probabilistic models into generalized field models. We argued why want to keep subcategorization and word order models separate in our approach, making it similar to MetaGrammar-like approaches, and we showed how we can link information about subcategorization and word order constraints to create the canonical linearization of a subcategorization frame for a particular word. As applications of our word order models we discussed their use in a computational typology of word order freedom, and in the construction of multilingual grammars.

Issues for further research concern (among others) the use of profile HMMs in *parsing*, and in *automatic test-suite construction*. Regarding the latter, we have developed an algorithm that can reformulate a generalized field model as a collection of set-based constraints (cf. (Duchier, 1999)) and then use these constraints to construct *negative examples*, i.e. linearizations that violate one or more word order restrictions. Because a purely data-driven approach to test-suite construction would be inherently biased by the kind of data one initially uses, we need some human intervention to correct and enhance the constraint-based models we obtain. It is at this point that we want to try to exploit the similarities between 3PGL and MetaGrammar, and to integrate the perspective on test-suite building presented in (Kinyon and Rambow, 2003): Kinyon and Rambow suggest to use MetaGrammars to automatically generate test-suites, thus minimizing human intervention (read: minimizing the cost of manually constructing a test-suite) – i.e. the kind of grammars that Kinyon and Rambow still build manually, but that we could (at least partly) learn from treebanks using 3PGL.

Acknowledgments

The work reported in this paper is supported by the DFG SFB 378 *Resource-Sensitive Cognitive Processes*, Project NEGRA EM6. The author would like to thank Petr Homola, for discussions and his work on implementing the profile HMM Java API used in this paper, as well as Peter Dienes, Detlef Prescher, and Karin Müller.

References

Jason Baldridge. 2002. *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh, Edinburgh, Scotland.

Marie-Hélène Candito. 1996. A principle-based hierarchical presentation of LTAGs. In *Proceedings COLING'96*, Copenhagen.

Eugene Charniak. 1996. Tree-bank grammars. In *Proceedings AAAI'96*, pages 1031–1036.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings NAACL-2000*, Seattle WA.

Michael Collins. 1997. Three generative models for statistical parsing. In *Proceedings ACL'97*.

Michael Collins. 1999. *Head-driven Statistical Models for Natural Language Processing*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

Amid Dubey and Frank Keller. 2003. Probabilistic parsing for German using sister-head dependencies. Under submission.

Denys Duchier and Ralph Debusmann. 2001. Topological dependency trees: A constraint-based account of linear precedence. In *Proceedings ACL'01*, Toulouse, France.

Denys Duchier. 1999. Axiomatizing dependency parsing using set constraints. In *Sixth Meeting on Mathematics of Language*, pages 115–126, Orlando, Florida, July.

Richard Durbin, Sean Eddy, Anders Krogh, and Graeme Mitchison. 1998. *Biological Sequence Analysis: Probabilistic models of proteins and nucleic acids*. Cambridge University Press.

Sisay Fissaha, Daniel Olejnik, Ralf Kornberger, Karin Müller, and Detlef Prescher. 2003. Experiments in German treebank parsing. In *Proceedings TSD'03*. Springer-Verlag.

Jan Hajič. 1998. Building a syntactically annotated corpus: The Prague Dependency Treebank. In Eva Hajičová, editor, *Issues of Valency and Meaning*, pages 106–132. Karolinum, Prague, Czech Republic.

Tilmann Höhle. 1983. *Topologische Felder*. Ph.D. thesis, University of Cologne.

Frank Keller. 2000. *Gradience in grammaticality*. Ph.D. thesis, University of Edinburgh, Edinburgh, Scotland.

Alexandra Kinyon and Carlos Prolo. 2002. A classification of grammar development strategies. In *Proceedings COLING'02*, Taipei.

Alexandra Kinyon and Owen Rambow. 2003. The metagrammar: A cross-framework and cross-language test-suite generation tool. In *Proceedings LINC'03*, Budapest.

Geert-Jan M. Kruijff. 2001. *A Categorial-Modal Logical Architecture of Informativity: Dependency Grammar Logic & Information Structure*. Ph.D. thesis, Charles University, Prague, Czech Republic.

Daniele Kurz, Wojciech Skut, and Hans Uszkoreit. 2000. German factors constraining word order variation. In *Thirteenth Annual Conference on Human Sentence Processing CUNY 2000*, La Jolla, California.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. In *Proceedings of the ARPA Human Language Technology Workshop*.

Nelleke Oostdijk. 2000. The spoken dutch corpus. overview and first evaluation. In *Proceedings LREC 2000*, pages 887–894.

Gerald Penn. 1999. A generalized-domain-based approach to Serbo-Croation second position clitic placement. In Gosse Bouma, Erhard Hinrichs, Geert-Jan M. Kruijff, and Richard T. Oehrle, editors, *Constraints and Resources in Natural Language Syntax and Semantics*, pages 119–136. CSLI Press, Stanford CA.

Wiebke Petersen. 2001. A set-theoretical approach for the induction of inheritance hierarchies. *Electronic Notes in Theoretical Computer Science*, 51.

Lawrence E. Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February.

Eric Sven Ristad. 1998. Hidden Markov Models with finite state supervision. In Andras Kornai, editor, *Extended Finite State Models of Language*. Cambridge University Press, Cambridge, England.

Anoop Sarkar and Daniel Zeman. 2000. Automatic extraction of subcategorization frames for czech. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, Saarbrücken, Germany.

Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. 1997. An annotation scheme for free word order languages. In *Applied Natural Language Processing 1997*, pages 88–95.

Caroline Sporleder. 2002. Learning lexical inheritance hierarchies with maximum entropy models. In *Wsh on Machine Learning Approaches in Computational Linguistics*, Trento, Italy.

Aline Villavicencio. 2002. *The Acquisition of a Unification-Based Generalised Categorial Grammar*. Ph.D. thesis, University of Cambridge, Cambridge, United Kingdom.

Fei Xia, Martha Palmer, and Aravind Joshi. 2000. A uniform method of grammar extraction and its application. In *Proceedings of EMNLP-2000*, Hong Kong.

A Profile HMMs: Theory

Profile HMMs are linear models that consist of *match, insert* and *delete* states. Match states represent the profile, insert states are used to match additional symbols inserted into sequences and delete states are used if a part of the profile is not present in a sequence. Whereas match and insert states generate emission symbols, delete states are *silent*, i.e. they do not generate any output. The length of a pHMM M is the number of its match states. Alike HMMs, we need to solve three basic problems for pHMMs, cf. (Rabiner, 1989):

Evaluation We are given a model M and a sequence of emission symbols $\alpha = x_1 \dots x_L$. We need to compute $P(\alpha|M)$, the probability that the sequence α has been generated by M.

Explanation We are given a model M and a sequence of emission symbols $\alpha = x_1 \dots x_L$ and need to compute the most probable state sequence, which generates the sequence α.

Parametrization We have a model M and a set of sequences of emission symbols $T = \{\alpha_i | \alpha_i = {x_1}^i \dots {x_L}^i\}$, and we need to adjust the parameters of M to maximize the likelihood of T.

The definitions we provide below are repeated from (Durbin et al., 1998).

Evaluation computes the probability of a sequence $\alpha = x_1 \dots x_L$ given the model M, i.e. the sum of probabilities of all state sequences that can generate α. This probability can be seen as a score of how well the model M matches the sequence α. We can compute this probability efficiently using a version of the *forward algorithm* adapted for the three types of states we have in pHMMs (M_k=match state k, I_k=insert state k, D_k=delete state k):

Initialization

$f_{M_0} = 1$

Recursion

$$\begin{aligned}
f_{M_k}(i) &= e_{M_k}(x_i)[f_{M_{k-1}}(i-1)a_{M_{k-1}M_k} + \\
&\quad + f_{I_{k-1}}(i-1)a_{I_{k-1}M_k} + \\
&\quad + f_{D_{k-1}}(i-1)a_{D_{k-1}M_k}] \\
f_{I_k}(i) &= e_{I_k}(x_i)[f_{M_k}(i-1)a_{M_k I_k} + \\
&\quad + f_{I_k}(i-1)a_{I_k I_k} + \\
&\quad + f_{D_k}(i-1)a_{D_k I_k}] \\
f_{D_k}(i) &= f_{M_{k-1}}(i)a_{M_{k-1}D_k} + \\
&\quad + f_{I_{k-1}}(i)a_{I_{k-1}D_k} + \\
&\quad + f_{D_{k-1}}(i)a_{D_{k-1}D_k}
\end{aligned}$$

Termination

$$
\begin{aligned}
f_{M_{M+1}}(L+1) &= f_{M_M}(L)a_{M_M M_{M+1}} + \\
&+ f_{I_M}(L)a_{I_M}a_{I_M M_{M+1}} + \\
&+ f_{D_M}(L)a_{D_M M_{M+1}}
\end{aligned}
$$

The value $f_X(i)$ gives the probability that the computation is in state S after reading $x_1 \dots x_i$, i.e. the first i symbols of the observed sequence. The value $f_{M_{M+1}}(L)$ is the probability of the whole sequence.

Explanation computes the most probable state sequence that would generate α, given a model M, and returns the probability of this state sequence. To find the state sequence, we use an adaptation of the Viterbi algorithm. The formulas are similar to the forward algorithm, except that we use $\max$ instead of the sum of probabilities.

Initialization

$f_{M_0} = 1$

Recursion

$$
\begin{aligned}
p_{M_k}(i) &= \max \begin{cases} e_{M_k}(i)p_{M_{k-1}}(i-1)a_{M_{k-1}M_k} \\ e_{I_k}(i)p_{I_{k-1}}(i-1)a_{I_{k-1}M_k} \\ p_{D_{k-1}}(i-1)a_{D_{k-1}M_k} \end{cases} \\
p_{I_k}(i) &= \max \begin{cases} e_{M_k}(i)p_{M_k}(i-1)a_{M_k I_k} \\ e_{I_k}(i)p_{I_k}(i-1)a_{I_k I_k} \\ p_{D_k}(i-1)a_{D_k I_k} \end{cases} \\
p_{D_k}(i) &= \max \begin{cases} p_{M_{k-1}}(i)a_{M_{k-1}D_k} \\ p_{I_{k-1}}(i)a_{I_{k-1}D_k} \\ p_{D_{k-1}}(i)a_{D_{k-1}D_k} \end{cases}
\end{aligned}
$$

Termination

$$
p_{M_{M+1}}(L+1) = \max \begin{cases} p_{M_M}(L)a_{M_M M_{M+1}} \\ p_{I_M}(L)a_{I_M}a_{I_M M_{M+1}} \\ p_{D_M}(L)a_{D_M M_{M+1}} \end{cases}
$$

Finally, **parametrization** adjusts the model parameters to the observed (training) sequences. In an unsupervised learning scenario, we can use the iterative forward-backward (Baum-Welch) algorithm to maximize the likelihood. The algorithm uses forward probabilities as defined above, and backward probabilities, as defined below:

Initialization

$$
\begin{aligned}
b_{M_{M+1}}(L+1) &= 1 \\
b_{M_M}(L) &= a_{M_M M_{M+1}} \\
b_{I_M}(L) &= a_{I_M M_{M+1}} \\
b_{D_M}(L) &= a_{D_M M_{M+1}}
\end{aligned}
$$

Recursion

$$
\begin{aligned}
b_{M_k}(i) &= e_{M_{k+1}}(x_{i+1})b_{M_{k+1}}(i+1)a_{M_k M_{k+1}} + \\
&+ e_{I_k}(x_{i+1})b_{I_k}(i+1)a_{M_k I_k} + \\
&+ b_{D_{k+1}}(i)a_{M_k D_{k+1}} \\
b_{I_k}(i) &= e_{M_{k+1}}(x_{i+1})b_{M_{k+1}}(i+1)a_{I_k M_{k+1}} + \\
&+ e_{I_k}(x_{i+1})b_{I_k}(i+1)a_{I_k I_k} + \\
&+ b_{D_{k+1}}(i)a_{I_k D_{k+1}} \\
b_{D_k}(i) &= e_{M_{k+1}}(x_{i+1})b_{M_{k+1}}(i+1)a_{D_k M_{k+1}} + \\
&+ e_{I_k}(x_{i+1})b_{I_k}(i+1)a_{I_k D_{k+1}} + \\
&+ b_{D_{k+1}}(i)a_{D_k D_{k+1}}
\end{aligned}
$$

The value $b_S(i)$ gives the probability that, starting in the state S, the sequence $x_{i+1} \dots x_{M+1}$ will be generated, i.e. the tail of the observed sequence of length $M - i$. The value $b_{M_0}(0)$ is the probability of the whole sequence.

The Baum-Welch algorithm consists of five steps:

1. Initialize the model parameters.
2. Compute forward and backward probabilities.
3. Compute relative counts.
4. Reestimate the model parameters.
5. Go to step 2, until the stop condition holds.

The relative counts are computed from forward and backward probabilities:

$$
\begin{aligned}
E_{X_k}(a) &= \tfrac{1}{P(x)} \textstyle\sum_{i|x_i=a} f_{X_k}(i)b_{X_k}(i) \\
A_{X_k M_{k+1}} &= \tfrac{1}{P(x)} \textstyle\sum_i f_{X_k}(i)a_{X_k M_{k+1}} \\
&\quad e_{M_{k+1}}(x_{i+1})b_{M_{k+1}}(i+1) \\
A_{X_k I_k} &= \tfrac{1}{P(x)} \textstyle\sum_i f_{X_k}(i)a_{X_k I_k}e_{I_k}(x_{i+1})b_{I_k}(i+1) \\
A_{X_k D_{k+1}} &= \tfrac{1}{P(x)} \textstyle\sum_i f_{X_k}(i)a_{X_k D_{k+1}}b_{D_{k+1}}(i)
\end{aligned}
$$

To reestimate the parameters, the relative counts are normalized (Σ is the set of emission symbols, S is the set of states)

$$
e'_X(a) = \frac{E_X(a)}{\sum_{x \in \Sigma} E_X(x)}, \; a'_{XY} = \frac{A_{XY}}{\sum_{Z \in S} A_{XZ}(x)}
$$

As the algorithm is guaranteed to find a local maximum, the difference between the old and reestimated parameters gets smaller during the iterative process. The stop condition is true, when the differences $|e'_X(a) - e_X(a)|$ and $|a'_{XY} - a_{XY}|$ are below a fixed threshold for all parameters.

Modularity of grammatical constraints in HPSG-based grammar implementations

Detmar Meurers **Kordula De Kuthy** **Vanessa Metcalf**
Department of Linguistics, The Ohio State University
{dm,kdk,vmetcalf}@ling.osu.edu

Abstract

This paper is a contribution to the discussion of the choices involved in implementing HPSG-based grammars and their consequences on the modularity and reusability of grammatical resources, a central issue in multi-lingual grammar development. Based on two examples from the English Resource Grammar (Flickinger et al., 2000), the treatment of unbounded dependencies and the analysis of optional arguments, we show that adding recursive relations to the expressive means available to the grammar writer results in more modular, transparent, and compact grammars. Given the frequent use of recursive relations in HPSG linguistics, grammar implementations integrating such relations can also be closer to the linguistic generalizations they intend to implement.

1 Introduction

The organization of a grammar in layered, reusable structures is a key methodological issue for multi-lingual grammar development, and arguably for sustainable grammar implementation efforts in general. This insight is reminiscent of the development in computer science, where as a result of the rise of software engineering in the 70s (Naur and Randell, 1968; Parnas, 1975) modular software written in high-level programming languages replaced the low-level coding of the early years. The abstraction and data encapsulation possibilities of high-level languages are viewed as essential to obtain reliable, maintainable and reusable software modules. As a result there is general agreement that efficiency should not be sought by coding at a low level but by intelligent compilation from such a high-level language to executable code.

In this paper, we want to contribute to a discussion of the expressive means of grammar implementation systems and how they can support the formulation of modular grammatical constraints that are reusable across languages. More specifically, we want to investigate the usefulness of recursive relational constraints for the implementation of HPSG-based grammars under this perspective. We base our discussion on two examples, the encoding of the treatment of unbounded dependencies and the analysis of optional complements, two key components of the English Resource Grammar (ERG, Flickinger et al., 2000) as the largest HPSG-based grammar for English currently available. We contrast the ERG encoding of these two issues with encodings that make use of recursive relations, such as the append or union relations frequently used in HPSG linguistics.[1]

2 Example 1: Unbounded dependencies

The English Resource Grammar (ERG) developed by the LinGO[2] project is a freely available broad-coverage, HPSG-based grammar of

[1] Relational goals in HPSG linguistics are often written in functional notation, e.g., $\boxed{1} \oplus \boxed{2}$ instead of append($\boxed{1}$,$\boxed{2}$,$\boxed{3}$).

[2] Cf., http://lingo.stanford.edu/

English, which is implemented in the LKB system (Copestake and Flickinger, 2000). The grammar contains a wealth of analyses of English phenomena, including a coverage of unbounded dependencies that is based on the proposal in Bouma et al. (2001) (henceforth: BMS). This proposal is particularly interesting under an implementation perspective since it replaces the need for computationally problematic empty elements with a lexical specification of ordinary, visible elements. The so-called SLASH amalgamation constraint ensuring this lexical specification is shown figure 1.

$$word \Rightarrow \begin{bmatrix} \text{LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{DEPS} \left\langle [\text{SLASH}\ \boxed{1}], \ldots, [\text{SLASH}\ \boxed{m}] \right\rangle \\ \text{BIND}\ \boxed{0} \end{bmatrix} \end{bmatrix} \\ \text{SLASH} \quad \left(\boxed{1} \cup \ldots \cup \boxed{m}\right) - \boxed{0} \end{bmatrix}$$

Figure 1: SLASH amalgamation as defined in Bouma et al. (2001, p. 20)

This constraint on words collects the information about dependents of that word which are not locally realized. More concretely, the SLASH value of each of the dependents on the DEPS list is collected and the SLASH value of a word is specified to be the union of the collected values (minus the value of BIND, which in the following is ignored). To express this generalization, the SLASH amalgamation principle of BMS in figure 1 makes use of the recursive relations set union ($\cup$) and set complement ($-$), as well as the use of $\boxed{1}, \ldots, \boxed{m}$ to express a relation accessing the SLASH value of every element of the DEPS list to be unioned into the SLASH value of the word.

In an implementation platform that supports relational constraints, the SLASH amalgamation principle can be directly expressed. For example, figure 2 shows how one can encode SLASH amalgamation in the Trale system (Meurers et al., 2002), an extension of the ALE parsing and generation system (Carpenter and Penn, 1996).[3]
The relational goal `collect_slashes` collects the SLASH values of all elements of the DEPS list. Its definition in figure 3 specifies that it applies when the list of dependents `Deps` is known to be

```
word *> synsem:(loc:cat:deps:Deps,
                nonloc:slash:Slash)
        goal collect_slashes(Deps,Slash).
```

Figure 2: SLASH amalgamation in Trale

either an empty or a non-empty list. In the latter case, `coll_slashes_aux` peels off one dependent at a time and unions each `Slash` value to obtain the list of all slashes.[4]

```
collect_slashes(Deps,Slash) if
  when(Deps = (e_list;ne_list),
       coll_slashes_aux(Deps,Slash)).

coll_slashes_aux([],[]) if true.
coll_slashes_aux([(nonloc:slash:Slash)
                  |Deps], AllSlash) if
       collect_slashes(Deps,DepsSlash),
       union(Slash,DepsSlash,AllSlash).
```

Figure 3: Definition of `collect_slashes`

Since the ERG is implemented in the LKB system, which does not support relational goals, the SLASH amalgamation approach of BMS is encoded by unfolding the relation between the SLASH of a word and that of its dependents into all of the different ways in which a word in this grammar can select arguments. To be able to do this, one needs to know more about the grammar to be able to determine the maximal number of elements for which the recursive relation needs to be unfolded. For the ERG, the maximal number of arguments of a word is four, so that SLASH amalgamation for arguments can be encoded by five type constraints as shown in figure 4 on the next page.

Let us take a close look at what distinguishes the two encodings in terms of generality, modularity, and transparency:

First, the principle in figure 1 is more modular since it encodes the collection of the SLASH value from any number of dependents without requiring additional information about the specifics of the grammar. Unfolding the principle into a fixed number of disjunctive cases, on the other hand, is dependent on such additional knowledge, namely the maximal number of elements that SLASH needs to be amalgamated from, which

[3] This is essentially the encoding used in the German HPSG-based grammar (Meurers and De Kuthy, 2001) developed in Trale as part of the SFB 340 Project B8 (http://www.sfs.uni-tuebingen.de/hpsg/archive/sfb340-b4-b8/index_engl.html).

[4] Note that one could also use a difference list encoding for SLASH sets in order to avoid this call to the relation union, an issue which is orthogonal to the one we focus on in this paper.

```
basic_zero_arg := lex_synsem &
  [ LOCAL.ARG-S < >,
    NON-LOCAL [ SLASH 0-dlist ]].

basic_one_arg := canonical_synsem &
  [ LOCAL.ARG-S < [ NON-LOCAL [ SLASH #slash ]] >,
    NON-LOCAL [ SLASH #slash ]].

basic_two_arg := lex_synsem &
  [ LOCAL.ARG-S < [ NON-LOCAL [ SLASH [ LIST #smiddle,
                                        LAST #slast ]]],
                  [ NON-LOCAL [ SLASH [ LIST #sfirst,
                                        LAST #smiddle ]]] >,
    NON-LOCAL [ SLASH [ LIST #sfirst,
                        LAST #slast ]]].

basic_three_arg := lex_synsem &
  [ LOCAL [ ARG-S < [ NON-LOCAL [ SLASH [ LIST #smiddle2,
                                          LAST #slast ]]],
                    [ NON-LOCAL [ SLASH [ LIST #sfirst,
                                          LAST #smiddle1 ]]],
                    [ NON-LOCAL [ SLASH [ LIST #smiddle1,
                                          LAST #smiddle2 ]]] > ],
    NON-LOCAL [ SLASH [ LIST #sfirst,
                        LAST #slast ]]].

basic_four_arg := lex_synsem &
  [ LOCAL [ ARG-S < [ NON-LOCAL [ SLASH [ LIST #smiddle3,
                                          LAST #slast ]]],
                    [ NON-LOCAL [ SLASH [ LIST #sfirst,
                                          LAST #smiddle1 ]]],
                    [ NON-LOCAL [ SLASH [ LIST #smiddle1,
                                          LAST #smiddle2 ]]],
                    [ NON-LOCAL [ SLASH [ LIST #smiddle2,
                                          LAST #smiddle3 ]]] > ],
    NON-LOCAL [ SLASH [ LIST #sfirst,
                        LAST #slast ]]].
```

Figure 4: The five type constraint encoding SLASH amalgamation in the ERG[6]

in the ERG happens to be four.

Second, the fact that the principle of BMS can collect the SLASH of any number of elements also means that it is more general than the ERG encoding. Even with specific knowledge about the grammar, it is impossible to capture the full generality of the BMS proposal in the ERG since the key idea of that paper is to generalize over adjunct and argument extraction—but the number of adjuncts is not lexically bounded, so that it is impossible to unfold all potential instances of amalgamation. It is therefore not surprising that in the ERG only argument extraction is handled via SLASH amalgamation, not dependent extraction in general, as proposed by BMS.[7]

Third, a lack of generality and modularity of the ERG encoding derives from the fact that the ERG encoding employs five independent constraints having five different types as antecedent. So while the principle of figure 1 on the preceding page applies to all words and thus is dependent only on what the linguist or grammar writer decided to classify as a word in the grammar, the ERG unfolding of SLASH amalgamation depends on five separate classifications, one for each type of antecedent. As before, this is a loss of modularity since an understanding of the ERG encoding of SLASH amalgamation is dependent on knowing where five types are used in the specification of lexical entries in the grammar, whereas the original principle of figure 1 on the facing page only requires knowledge of where a single type, *word*, is used in the specification of lexical entries in the grammar.

Fourth, a further lack of generality and transparency of the ERG encoding is caused by the fact that the ERG encoding imposes five type constraints having five different consequents, whereas a recursive encoding consists of a base clause and a recursive clause characterizing what is the case at $n + 1$ based on knowledge of the state of affairs at n. The recursive case thus is a generalization over all cases, starting from the base case; with a recursive definition it is impossible for e.g. the fourth case to differ from the third case in any other way than exactly the way in which the third case differed from the second case. In contrast, there is no such generality across cases for five separately written down constraints, such as in the ERG encoding of SLASH amalgamation in figure 4. How transparent would it be if, e.g., a couple of variable names in `basic_tree_arg` would be changed such that the SLASH value of one of the three arguments is not collected? Interestingly, closer inspection of the ERG encoding reveals just such a non-generality across cases: the `basic_one_arg` case happens to be special in that it requires the word from which it collects slashes to have a `canonical_synsem`, whereas all other cases require a `lex_synsem`. There thus is a clear contrast to the SLASH amalgamation principle of BMS which generalizes over all cases. This generalization is not expressed in the ERG encoding; instead, one needs to look at every one of the five constraints separately to know what

[7]In languages exhibiting coherence or restructuring phenomena (e.g., German, Dutch, and the Romance languages), even the number of arguments is not bounded in the lexicon since under the normal HPSG analyses of those languages, certain verbs are specified to attract the arguments of their complement.

[7]The constraints are shown without the specification of the QUE and REL attributes. These are amalgamated as well, so that the actual ERG constraints are three times as large as the ones shown in figure 4.

exactly happens to be encoded in each one.

To address the gap in generality, modularity and transparency between the linguistic principle of BMS and the ERG encoding of it, the recursive relations used in the linguistic principle need to be supported by the grammar implementation system. To address all of the issues we raised above, including the unbounded number of potential dependents, recursive relations need to be fully supported by the system in the sense that the runtime environment must support the execution of recursive relations. The overhead associated with such a runtime support of relations can often be avoided though by unfolding and inlining the relation calls at compile time. This corresponds to inlining of functions and unfolding of loops as a standard option of compilers for many programming languages. The use of SLASH amalgamation made in the ERG is an instance where compile-time unfolding of the relation at compile-time is possible (unless SLASH amalgamation is also applied to adjuncts). Note that as long as the grammar is specified with a relation explicitly encoding the generalization to be captured, most of the shortcomings of the ERG encoding we discussed above do not apply, independent of whether the relational constraints are ensured by unfolding them at compile-time or by executing them at run-time.

3 Example 2: Optional complementation

The second example we want to look at in this paper concerns the analysis of optional complements in the ERG, which is also discussed in Flickinger (2000).[8] The empirical issue of verbs with optional complements is illustrated by the sentences in (1), which are licensed by the ERG.

(1) a. Kim bet Tom five dollars that they hired Cindy.
  b. Kim bet Tom five dollars.
  c. Kim bet Tom that they hired Cindy.
  d. Kim bet five dollars that they hired Cindy.
  e. Kim bet five dollars.
  f. Kim bet that they hired Cindy.
  g. Kim bet Tom.
  h. Kim bet.

[8]Whether the treatment of optional complements proposed in Flickinger (2000) is the best analysis for this phenomenon is an orthogonal issue. Our focus is on how the proposed analysis is reflected in the implementation.

In sentence (1a), the verb *bet* takes a subject *Kim* and three complements, the NPs *Tom* and *five dollars*, as well as the sentential complement *that they hired Cindy*. The other sentences in (1) exemplify that each of those three complements is optional.

The brute-force method for licensing these structures would be to posit eight independent lexical entries for *bet*, one for each of the environments exemplified above. But this would miss the generalization that *bet* has three complements, each of which can be realized or not. As discussed by Flickinger (2000), the ERG takes this generalization into account and posits only the single lexical entry shown in figure 5.[9]

$$\begin{bmatrix} \textit{non\_derived\_word} \\ \text{PHON}\ \langle \textit{bet} \rangle \\ \text{S} \begin{bmatrix} \textit{canonical\_synsem} \\ \text{L|C} \begin{bmatrix} \text{HEAD}\ \textit{verb} \\ \text{VAL} \begin{bmatrix} \text{COMPS} \left\langle \begin{bmatrix} \textit{synsem} \\ \text{L|C|H} \begin{bmatrix} \textit{nominal} \\ \text{CASE}\ \textit{acc} \end{bmatrix} \\ \text{OPT}\ \textit{plus} \end{bmatrix}, \begin{bmatrix} \textit{synsem} \\ \text{L|C|H} \begin{bmatrix} \textit{nominal} \\ \text{CASE}\ \textit{acc} \end{bmatrix} \\ \text{OPT}\ \textit{plus} \end{bmatrix}, \begin{bmatrix} \textit{synsem} \\ \text{L|C|H} \begin{bmatrix} \textit{verbal} \\ \text{VFORM}\ \textit{fin} \end{bmatrix} \\ \text{OPT}\ \textit{plus} \end{bmatrix} \right\rangle \\ \text{SUBJ} \left\langle \begin{bmatrix} \textit{synsem} \\ \text{L|C|H}\ \textit{nominal} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 5: Lexical entry for *bet*

The key aspect here is the specification of the complement requirements on the COMPS list. The list contains three elements, each of which is marked as optional with the help of an attribute OPT(IONAL) appropriate for *synsem* objects.

In figure 6 on the facing page we see the structure that is licensed for a sentence in which none of the optional complements are realized, i.e., sentence (1h). The entry of *bet* can construct as the head daughter of such a head_subject_phrase even though it has not yet realized its complements. This is possible since, different from the traditional HPSG analysis (Pollard and Sag, 1994), the head daughter is not required to be saturated, i.e., have a COMPS value of type *e_list*. Instead, a sign is also understood to be saturated for complements if it has only optional complement requirements left.

[9]Here and in the following figures, only the specifications relevant to the issue of optionality are shown. For space reasons, attributes are sometimes abbreviated by their first letter.

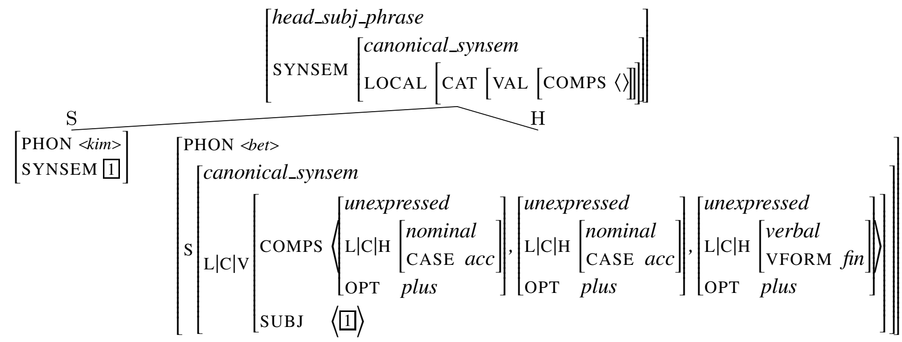


Figure 6: A sentence with three unrealized complements

Adding the head-complement phrase of the ERG to the picture, one can also license (1b) and (1g), which are sentences in which one or two complements are realized and the other complements, which are more oblique than the ones that are realized, are missing.[10] Figure 7 shows the relevant aspects of the definition of head-complement phrases in the ERG. Note that it is always the first element of the COMPS list that is realized as the *non_head_dtr* of such a phrase.

$$\textit{head\_comp\_phrase} \rightarrow \begin{bmatrix} \text{SYNSEM} \begin{bmatrix} \textit{canonical\_synsem} \\ \text{LOCAL|CAT|VAL|COMPS } \boxed{2} \end{bmatrix} \\ \text{HEAD\_DTR|SYNSEM} \begin{bmatrix} \textit{canonical\_synsem} \\ \text{LOCAL|CAT|VAL|COMPS } \langle \boxed{1} | \boxed{2} \rangle \end{bmatrix} \\ \text{NON\_HEAD\_DTR|SYNSEM } \boxed{1} \textit{canonical\_synsem} \end{bmatrix}$$

Figure 7: The realization of COMPS requirements in the head-complement rule of the ERG

Figure 8 on the following page shows the structure that the ERG assigns to the sentence (1g). The lower tree is an instance of a head_comp_phrase, in which the first subcategorization requirement on COMPS, namely the NP *Tom* bearing the tag $\boxed{2}$, is realized. The *head_subject_phrase* on top is licensed just as in the previous example, marking the remaining optional elements on the COMPS list of the head daughter *bet Tom* as unexpressed.

Since the *head_comp_phrase* in the ERG always realizes the first element of the COMPS list, a problem arises if one wants to license a sentence in which the least oblique complement, i.e., the first element on the COMPS list is optional and not realized. Note that this is not an accidental oversight in the formulation of the rule licensing *head_comp_phrases* in the ERG; rather it is a consequence of the fact that the LKB system does not support relational goals as attachment to phrase structure rules. We will see in the next section that an implementation platform that includes such relational goals can express the relevant generalization, namely that the head_comp_phrase realizes the first requirement on COMPS which is not marked as unrealized optional element. In the ERG as implemented in the LKB system, the problem is addressed by introducing additional types of phrases which eliminate the unrealized optional subcategorization requirements from the front of the COMPS list in order to bring the requirement intended to be realized to the first position of the COMPS list. For this purpose, in addition to the ordinary *head_comp_phrases*, the ERG needs two additional rules: the *head_opt_comp_phrases* which eliminates one optional complement from the front of the COMPS list, and the *head_opt_two_comp_phrases* which eliminate first two complement requirements from the COMPS list. Further additional phrases would be needed if the grammar had COMPS lists longer than three.

Figure 9 on the next page illustrates the structure licensed for sentence (1e), in which only the second most oblique complement is realized. The unary structure at the bottom of the tree is an instance of the additional *head_opt_comp_phrase*, whose purpose is the elimination of the first com-

[10]The COMPS is ordered by obliqueness, with the least oblique complement being the first element of the list.

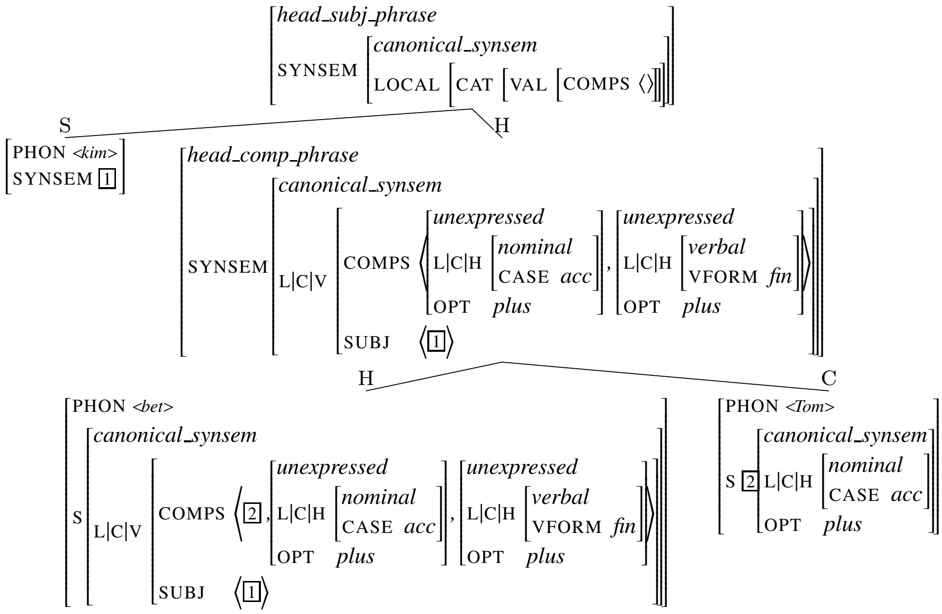


Figure 8: A sentence in which the two most oblique complements are not realized

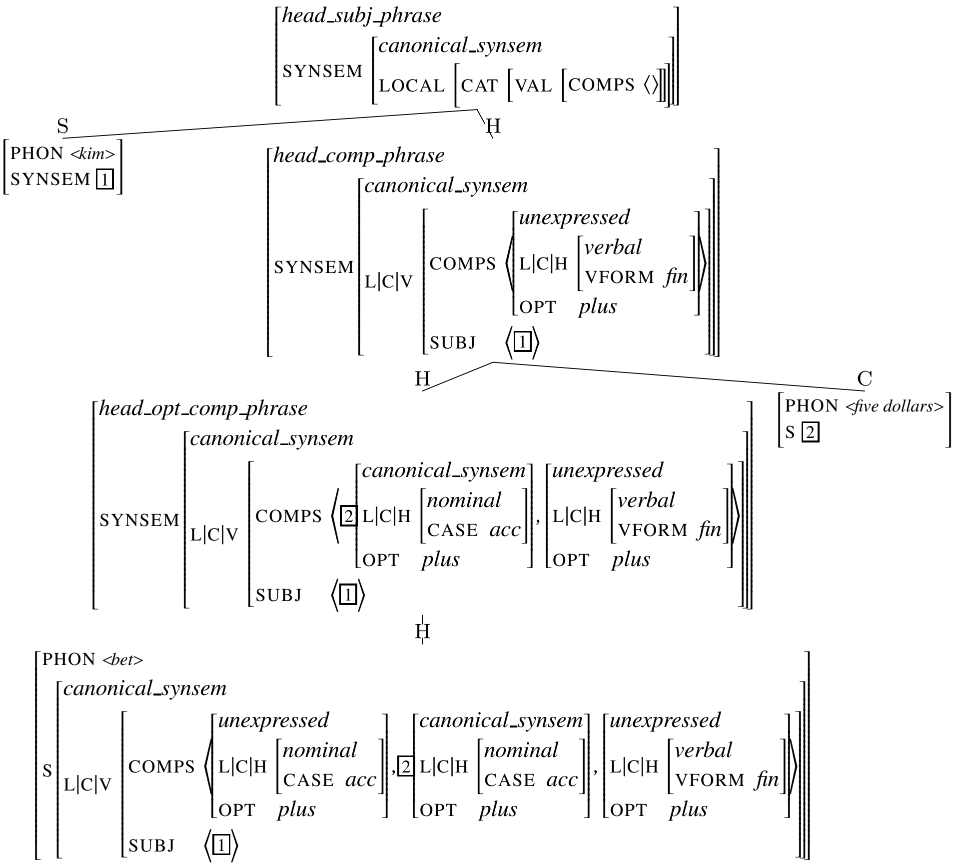


Figure 9: The ERG analysis of a sentence in which only the second most-oblique object is realized

plement requirement, an unexpressed optional object NP, in order to bring the requirement 2 to the front of the COMPS list. That complement (*five dollars*) is then realized in the *head_comp_phrase* dominating the *head_opt_comp_phrase*.

Capturing the missed generalization We saw above that the ERG analysis of optional complements requires three different head-complement rules since in the LKB system there is no way to express the relevant generalization that one wants to realize the first element on the comps list that is not an unrealized optional argument.

The revised head_complement rule in figure 10 shows how the intended generalization can be expressed using an append relation (⊕) to state that the element $\boxed{1}$ to be realized can be preceded by an *o_list*, the type used in the ERG to refer to a list of unrealized optional elements. In a grammar

$$\textit{head\_comp\_phrase} \rightarrow \left[\begin{array}{l} \text{SYNSEM}\left[\begin{array}{l}\textit{canonical\_synsem}\\ \text{LOCAL|CAT|VAL|COMPS}\ \boxed{2}\end{array}\right]\\ \text{HEAD\_DTR|S}\left[\begin{array}{l}\textit{canonical\_synsem}\\ \text{LOCAL|CAT|VAL|COMPS}\ \textit{o\_list} \oplus \langle \boxed{1}|\boxed{2}\rangle\end{array}\right]\\ \text{NON\_HEAD\_DTR|SYNSEM}\ \boxed{1}\textit{canonical\_synsem}\end{array}\right]$$

Figure 10: Generalized realization of COMPS requirements in the revised head-complement rule

including this revised *head_complement_phrase* instead of the original one from the ERG we saw in figure 7 on page 5, the types and definitions for *head_opt_comp_phrases* and *head_opt_two_comp_phrases* are no longer needed. Interestingly, the LKB encoding of the ERG using a *head_complement_phrase* plus the two 'auxiliary' phrase types used to unearth the first complement requirement to be realized can be seen as the result of unfolding the first three calls to the append (⊕) relation in the revised *head_complement_phrase* defined in figure 10, i.e., the LKB encoding can result from a compilation step taking the more general encoding as its input. This means that the issue of enabling the grammar writer to express the full generalization in the grammar as shown by the revised encoding is independent of the as yet unresolved question of the relative efficiency of parsing systems with and without runtime support for relational goals.

4 Summary and Outlook

In this paper we discussed the use of relational constraints in the implementation of HPSG-based grammar in pursuit of a modular and reusable grammar encoding. We based the discussion on two examples from the English Resource Grammar, the largest HPSG-based grammar for English currently available, and an insightful collection of analyses of many aspects of English syntax.

In the first example, we showed that the ERG unfolds the general SLASH amalgamation principle of Bouma et al. (2001) into the specific cases assumed for this particular grammar, which is necessary since relational goals are not supported in the LKB system. The resulting encoding does not capture the full generality of the principle and is less modular and transparent than the formulation of Bouma et al. (2001) or its computational encoding in a framework incorporating recursive relations, such as the Trale system.

In the second example, we discussed how the ERG captures the optionality of arguments through the use of a single lexical entry, coupled with an ontology of markings distinguishing optional from obligatory and unrealized from realized elements. Subject-head and head-complement structures are modified accordingly, but due to the lack of a possibility to use recursive relations in grammars implemented in the LKB system, the ERG analysis fails in treating optional arguments in a general way, requiring two new types of 'auxiliary' phrases which are otherwise unmotivated. The focus on a very lean system without relational goal attachments to phrase structure rules thus results in a loss of generality and thereby transparency and modularity of the grammars that can be expressed. Apart from the software engineering aspect, this also breaks the clear link of the implementation to linguistic theory, which in Copestake and Flickinger (2000) is identified as the hallmark distinguishing the ERG from other grammar implementation efforts such as those around the Alvey Natural Language Tools (Briscoe et al., 1987).

We showed that a recoding of the analysis of optionality in a system supporting relational attachments can overcome this shortcoming by making use of a single recursive relation, append, used to select the first non-optional argument on a list. A system including relational attachments thus is better suited to achieve the goal of a modular and linguistically informed grammar implementation.

We are in the process of reimplementing the English Resource Grammar to also investigate in practice which expressive means are useful for writing modular and transparent HPSG-based grammars. The Milca English Resource Grammar (MERGE, De Kuthy et al., 2003) is implemented in the Trale system (Meurers et al., 2002), an extension of the ALE parsing and generation system (Carpenter and Penn, 1996). In line with the arguments presented in Götz and Meurers (1997) and standard practice in HPSG linguistics, the Trale system supports both implicational constraints and relational goals such as the ones used in the approach to unbounded dependencies of Bouma et al. (2001) and the analysis of optional arguments of Flickinger (2000) which were discussed in this paper.

References

Gosse Bouma, Rob Malouf and Ivan A. Sag, 2001. Satisfying Constraints on Extraction and Adjunction. *Natural Language and Linguistic Theory*, 19(1):1–65.

Ted Briscoe, Claire Grover, Bran Boguraev and John Carroll, 1987. A formalism and environment for the development of a large grammar of English. In *Proceedings of IJCAI-87*. Milan.

Bob Carpenter and Gerald Penn, 1996. Compiling Typed Attribute-Value Logic Grammars. In Harry Bunt and Masaru Tomita (eds.), *Recent Advances in Parsing Technologies*, Kluwer, Dordrecht, pp. 145–168.

Ann Copestake and Dan Flickinger, 2000. An open source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-00)*, Athens.

Kordula De Kuthy, Vanessa Metcalf and W. Detmar Meurers, 2003. The Milca English Resource Grammar (MERGE). Trale grammar code and documentation. Department of Linguistics. The Ohio State University. To be released in Dec. 2003; for preliminary versions cf. http://ling.osu.edu/~dm/merge/.

Dan Flickinger, 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28.

Dan Flickinger, Ann Copestake and Ivan A. Sag, 2000. HPSG Analysis of English. In Wolfgang Wahlster (ed.), *Verbmobil: Foundations of Speech-to-Speech Translation*, Springer, Berlin, Artificial Intelligence, pp. 254–263.

Thilo Götz and W. Detmar Meurers, 1997. Interleaving Universal Principles and Relational Constraints over Typed Feature Logic. In *Proceedings of the 35th Annual Meeting of the ACL and 8th Conference of the EACL*. Madrid, pp. 1–8. http://ling.osu.edu/~dm/papers/acl97.html.

W. Detmar Meurers and Kordula De Kuthy, 2001. Case Assignment in Partially Fronted Constituents. In Christian Rohrer, Antje Roßdeutscher and Hans Kamp (eds.), *Linguistic Form and its Computation*, CSLI Publications, Stanford, CA, pp. 29–63.

W. Detmar Meurers, Gerald Penn and Frank Richter, 2002. A Web-based Instructional Platform for Constraint-Based Grammar Formalisms and Parsing. In *Effective Tools and Methodologies for Teaching NLP and CL*. Proceedings of the Workshop held at 40th Annual Meeting of the ACL. Philadelphia, PA.

Peter Naur and Brian Randell (eds.), 1968. *Software Engineering: Report on a conference sponsored by the NATO Science Committee, Garmisch-Partenkirchen, Germany, 7–11 Oct. 1968*. Science Affairs Division, Nato, Brussels.

David Lorge Parnas, 1975. Software Engineering or Methods for the Multi-Person Construction of Multi-Version Programs. In Clemens Hackl (ed.), *Programming Methodology, 4th Informatik Symposium, IBM Germany, Wildbad, September 25-27, 1974*, Springer Verlag, Lecture Notes in Computer Science, pp. 225–235.

Carl Pollard and Ivan A. Sag, 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press and CSLI Publications.

LKB grammar development: French and beyond

Jesse Tseng
MoDyCo (UMR 7114)
Université Paris X – Nanterre
`jesse.tseng@linguist.jussieu.fr`

Abstract

This paper presents some aspects of an HPSG-style grammar for French currently under development using the LKB platform, with particular emphasis on a number of technical solutions and linguistic analyses that may be relevant to the implementation of constraint-based grammars of other languages. The main issues discussed are lexical rule management, the implementation of argument composition, and the analysis of phrasal affixes.

The French grammar described in this paper has been in development since early 2001.[1] As is the case for many projects of this type, the grammar was built more or less from scratch, without direct reuse of existing grammar resources. It was however heavily influenced by the developers' previous experiences with other implementation projects for a variety of languages. The objective of this paper is to give a general overview of the current state of the grammar, focusing on a number of specific issues of wider applicability, beyond the monolingual treatment of French.

1 General presentation

We use the LKB platform for typed feature structure-based grammar development (Copestake, 2002). In this discussion we assume that the reader is familiar with this platform and the necessary distinction between a theoretical HPSG grammar (with its almost abitrarily expressive description language) and an HPSG-style grammar implemented with the LKB (or any other development platform). Specific instances of this divergence include: the unavailability of disjunction/negation, the absence of constraints with complex antecedents, the need to specify explicitly the number of daughters in every rule, and to fix their linear order. None of these issues will be addressed here.

In the initial stages, we have concentrated our efforts on a subset of grammatical phenomena of linguistic interest, rather than striving for wide coverage of constructions and lexical items. The grammar currently provides a treatment of the following phenomena of (written) French:

- verbal conjugation and complementation classes
- realization of bound pronominal clitics as verbal affixes (Miller and Sag, 1997), or as independent syntactic words
- argument extraction for interrogatives and relative clauses(Bouma et al., 2001)
- auxiliary and causative constructions allowing "clitic climbing" (Abeillé and Godard, 1997; Abeillé and Godard, 2002)
- past participle agreement (Abeillé and Godard, 1996)

[1] Most of the work reported here was carried out in the research groups TALaNa/Lattice (UMR 8094) and LLF (UMR 7110) at Université Paris 7. The author wishes to thank the members of these groups for fruitful collaboration.

- orthographic alternations reflecting elision, contraction, and liaison (Tseng, 2003b; Tseng, 2003a)

As indicated in the list above, the grammar incorporates the insights of a great deal of theoretical work on the formal analysis of French in HPSG. In fact, the main objective of the grammar at this stage is to evaluate the technical adequacy and the successful interaction of recent theoretical proposals. When a choice between efficiency and theoretical adequacy presents itself, therefore, we generally decide in favor of the more linguistically justified analysis. For example, strict binary branching is a useful constraint on processing, but empirical evidence points to a flat structure for complex VPs in French, and so this is the analysis the implementation produces.

We will discuss three main topics in this talk: our approach to lexical rule management, some problems associated with the implementation of argument composition (for auxiliary and causative constructions), and the analysis of certain "weak" elements as phrasal affixes.

2 Lexical rule application

In many cases, the formal linguistic proposals that the grammar is based on can only be implemented in modified form, since the LKB system does not fully support the considerable expressive power of the HPSG formalism, and because the underlying type logic of the LKB differs from that commonly assumed by HPSG linguists. Consequently, certain constraints that are easily stated in the HPSG description language require a more complex procedural implementation. In our grammar, this is especially apparent at the lexical level: in order to construct the rich lexical descriptions needed to drive the syntactic derivation, the grammar makes use of a large network of lexical rules, most of which have no theoretical counterpart.[2]

[2]In a grammar incorporating the recent construction-based proposals for HPSG (e.g., Ginzburg and Sag (2001)), similar effects would be seen at the phrasal level, with a multiplication of non-branching syntactic rules. At present, the French grammar makes no use of phrasal subtypes.

2.1 Rule ordering

We adopt the partition of lexical objects into the subtypes *lexeme* and *word*; only words can be input to syntactic rules. This allows the divison of lexical rules into three types (*lexeme-to-lexeme*, *lexeme-to-word*, and *word-to-word*) with a rudimentary built-in ordering: *l2l* rules apply first, followed by exactly one *l2w* rule, and then any *w2w* rules. This means that *l2w* rules do not need to be relatively ordered, and they can never apply recursively. For *l2l* and *w2w* lexical rules, on the other hand, both of these problems have to be dealt with.

As an example, consider the inflection of French adjectives for gender and number. In the lexicon, adjectival entries are of type *lexeme*, and they have underspecified GEN and NUM values. These lexemes must undergo GEN instantiation (by one of two *l2l* lexical rules) followed by NUM instantiation (by one of two *l2w* lexical rules). Iteration of the *l2l* gender instantiation rules has to be blocked, for example by adding an ad hoc feature whose value changes so that the output no longer unifies with the input.

This is a very common situation, and so instead of inventing a new feature for every set of *l2l* rules, we use a general list-valued attribute STACK (appearing only on lexemes) to manage lexical rule application. Adjectival lexemes, for example, are listed in the lexicon with a single STACK element, linked to the (underspecified) INDEX value. The presence of this STACK element triggers either the masculine or the feminine GEN instantiation rule, which produces as output a lexeme with a new STACK list, containing just the (underspecified) NUMBER value. This STACK value triggers the singular (or plural) *l2w* NUM instantiation rule, which produces a word (with no STACK feature). The derivation of the feminine plural forms `grandes` is shown in Figure 1.

The STACK feature corresponds to nothing in theoretical HPSG, but it is a practical, transparent way of keeping track of the many "layers" of lexical rules. Technically, the STACK value is a list of objects of any type (including arbitrary strings), and every lexical rule looks for lexemes bearing a specific type of first STACK element. The rule modifies the STACK appropriately to send the output lexeme to the next lexical rule (or in the case

$$\begin{bmatrix} \textit{adj-lex} & \\ \text{ORTH} & \langle \texttt{"grand"} \rangle \\ \text{STACK} & \langle \boxed{1} \rangle \\ \text{INDEX} & \boxed{1} \begin{bmatrix} \text{NUM} & \boxed{2}\ \textit{num} \\ \text{GEN} & \textit{gen} \end{bmatrix} \end{bmatrix} \begin{array}{c} \textit{fem-lr} \\ \mapsto \\ (\textit{l2l}) \end{array} \begin{bmatrix} \textit{lexeme} & \\ \text{ORTH} & \langle \texttt{"grande"} \rangle \\ \text{STACK} & \langle \boxed{2} \rangle \\ \text{INDEX} & \boxed{1} \begin{bmatrix} \text{NUM} & \boxed{2}\ \textit{num} \\ \text{GEN} & \textit{fem} \end{bmatrix} \end{bmatrix} \begin{array}{c} \textit{plur-lr} \\ \mapsto \\ (\textit{l2w}) \end{array} \begin{bmatrix} \textit{word} & \\ \text{ORTH} & \langle \texttt{"grandes"} \rangle \\ \text{INDEX} & \boxed{1} \begin{bmatrix} \text{NUM} & \textit{plur} \\ \text{GEN} & \textit{fem} \end{bmatrix} \end{bmatrix}$$

Figure 1: STACK for adjective inflection

of an *l2w* rule, the output is a word and can appear in the syntax without further manipulation).

2.2 List recursion

The STACK is also useful in situations where *l2l* rules do have to apply iteratively for list processing, a very common operation in HPSG. For instance, argument realization in French involves the resolution of each *synsem* object on a verb's ARG-ST list to a specific subtype (*canon-*, *affix-*, or *gap-synsem* for canonically expressed, cliticized, and extracted arguments, respectively). In the LKB grammar, this resolution must be achieved explicitly, with a distinct lexical rule for each *synsem* subtype. We assume that verbal lexemes copy their arguments onto STACK, thereby triggering the application of one of the three kinds of argument realization lexical rule. Each rule resolves the type of the first *synsem* on the input STACK, and makes the appropriate changes to the lexical entry (i.e., the argument is added to the COMPS list, to the SLASH set, or to one of the clitic lists). The resolved *synsem* object is then popped off the STACK list in the output of the lexical rule. When the verb's STACK is empty, all of its arguments have been processed, and the verb passes on to the next set of lexical rules (e.g., clitic prefixation). An example of this procedure applied to the verb *donnent* is given in Figure 2.

Another example of STACK recursion is the treatment of French compound tenses (a form of the auxiliary *être* or *avoir* plus a part participle). The selection of the auxiliary and presence or absence of agreement inflection on the past participle are determined by inspection of the elements of the participle's ARG-ST list. If any of the arguments is reflexive, the the auxiliary must be *être* instead of *avoir*, and if there is a non-canonical (i.e., affix or gap) accusative argument, the participle must agree in gender and number with this argument (Abeillé and Godard, 1996). See (1c) below, for example. In the LKB implementation, again, this kind of list inspection requires a chain of recursive lexical rules. Past participles copy their ARG-ST list onto STACK and undergo a special set of STACK-popping lexical rules that determine auxiliary selection and agreement.

3 Underspecified arguments

For most verbs, the list of arguments in STACK is defined lexically (in direct correspondance to the ARG-ST list) and so the argument realization procedure described in the previous section will eventually terminate, once the STACK list is empty. A problem arises, however, in the treatment of the French temporal auxiliary verbs *être* and *avoir*, and the causative verb *faire*. These verbs have a lexically underspecified ARG-ST list, which is instantiated by the infinitival or participial V complement to the right (Abeillé et al., 1998; Abeillé and Godard, 2002). The result is a complex predicate construction with argument composition, an analysis that allows a straightforward treatment of the phenomenon of "clitic climbing," where a complement selected by a lower verb is realized as a pronominal clitic attached to a higher verb:

(1) a. Jean les voit. 'Jean sees them' (object realized as clitic on verb)

b. J veut les voir. 'J wants to see them' (no climbing)

c. Jean les a vus. 'J has seen them' (clitic climbing onto auxiliary)

$$\begin{bmatrix} \text{ORTH} & \langle\texttt{"donnent"}\rangle \\ \text{STACK} & \langle\boxed{3},\boxed{2},\boxed{1}\rangle \\ \text{ARG-ST} & \langle\boxed{1}\,[nom],\boxed{2}\,[acc],\boxed{3}\,[dat]\rangle \\ \text{VAL}\mid\text{COMPS} & \langle\,\rangle \end{bmatrix} \overset{canon\text{-}comp\text{-}lr}{\mapsto} \begin{bmatrix} \text{STACK} & \langle\boxed{2},\boxed{1}\rangle \\ \text{ARG-ST} & \langle\boxed{1},\boxed{2},\boxed{3}\ canon\text{-}ss\rangle \\ \text{VAL}\mid\text{COMPS} & \langle\boxed{3}\rangle \end{bmatrix}$$

$$\overset{gap\text{-}comp\text{-}lr}{\mapsto} \begin{bmatrix} \text{STACK} & \langle\boxed{1}\rangle \\ \text{ARG-ST} & \left\langle\boxed{1},\boxed{2}\begin{bmatrix} gap\text{-}ss \\ \text{LOC}\ \boxed{4} \end{bmatrix},\boxed{3}\right\rangle \\ \text{VAL}\mid\text{COMPS} & \langle\boxed{3}\rangle \\ \text{SLASH} & \{\boxed{4}\} \end{bmatrix} \overset{aff\text{-}subj\text{-}lr}{\mapsto} \begin{bmatrix} \text{STACK} & \langle\,\rangle \\ \text{ARG-ST} & \langle\boxed{1}\ aff\text{-}ss,\boxed{2},\boxed{3}\rangle \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \langle\boxed{1}\rangle \\ \text{COMPS} & \langle\boxed{3}\rangle \end{bmatrix} \\ \text{SLASH} & \{\boxed{4}\} \end{bmatrix}$$

Figure 2: STACK for argument realization

In our grammar, a verb with an underspecified ARG-ST list will also have an underspecified STACK. Like other verbs, auxiliary/causative verbs must undergo argument realization to determine VALENCE, SLASH, and cliticization properties. The problem is that the parser works from left to right, and so it tries to process the underspecified auxiliary/causative verb before identifying its participial/infinitival complement. The application of the STACK-popping argument realization rules to a variable STACK leads to infinite recursion.

One solution would be to modify the parsing procedure to delay lexical rule application until the STACK is instantiated, and meanwhile to let the parser look ahead to identify the following verbal complement.[3] A second possibility would be to prevent the application of the normal argument realization principles, but this is linguistically unmotivated, since auxiliary/causative verbs behave exactly like other verbs in this respect.

At present, we have implemented a lexical rule based solution that simply fixes the length of the underspecified STACK, from zero to four elements. This ensures termination, but it impairs the performance of the parser, because every occurrence of an auxiliary or causative verb triggers the generation of hundreds of chart edges. For any given STACK element, up to 8 different lexical rules can apply, so in principle the number of possible lexical entries generated is $8^0 + 8^1 + 8^2 + 8^3 + 8^4$. In fact, the actual number of entries is much lower, because not all rule sequences are allowed (for example, at most one argument can be extracted, and there are various constraints on clitic cooccurrence). Nevertheless, the grammar is significantly slower when processing any input involving argument inheritance. And it should be noted that the arbitrary upper limit of four inherited arguments on STACK is certainly too low.

This is a part of the implementation that calls for reexamination. The problem described here is by no means specific to French. The same mechanism of argument composition has been proposed for the corresponding constructions in the other Romance languages (Abeillé and Godard, in press), and for the treatment of non-finite complementation in German and Dutch (Hinrichs and Nakazawa, 1994; van Noord and Bouma, 1996; Meurers, 2000).

4 Phrasal affixes

The French LKB grammar incorporates a treatment of certain "weak form function words" (such as pronominal clitics) as bound morphological affixes, rather than syntactically independent words. In the case of clitics, this analysis is relatively easy to implement: an ARG-ST element of a verb, instead of being mapped to a VALENCE list, is speci-

[3] This is the approach adopted in the B8 grammar of German, implemented using Trale; see (Kordoni, 1999; Tseng, 2000) for some documentation. The possibility of "delaying" in the LKB should be explored.

fied as an *affix-synsem* and triggers the application of a lexical rule that realizes the appropriate prefix (or suffix) on the verb (Miller and Sag, 1997). This approach is possible because clitics are always arguments selected by another element, and they are always realized morphologically on this element (the verb).

We extend the affix analysis to another class of "weak" forms in French, including the definite article "le", and the prepositions "à" and "de", adopting and further elaborating the GPSG proposals of Miller (1992). In these cases, the analysis is less straightforward, because the prefix is a functor syntactically and semantically, with grammatical scope over an entire phrase. This is why authors like Miller call these elements "phrasal affixes." At the same time, however, their status as prefixes implies that they must be realized morphologically at the lexical level, attached to a single word. (Any approach based on post-syntactic affixation would be inconsistent with the lexicalist foundations of HPSG.)

4.1 Implementation

The key to the analysis is the dissociation of the morphological realization of the affix (at the lexical level) and the incorporation of its syntactic and semantic interpretation (in the syntax) (Tseng, 2003b). Concretely, a set of *word-to-word* lexical rules allows the prefixation of "le," "les," "de," etc. to more or less any existing word. This is where phenomena such as haplology, liaison/elision, and idiosyncratic contracted forms like "du" and "aux" are dealt with. At this point, the syntactic function and semantic contribution of the prefix are not yet activated. Instead, the presence of the prefix is encoded as a positive (left) EDGE feature that can only propagate from the left-most daughter (not necessarily the head) in a syntactic combination (non-peripheral daughters are constrained to have no positive EDGE specifications). In the syntax, when the appropriate phrase has been constructed, a unary syntactic rule can apply to add in the grammatical effects of the prefix, at the same time "discharging" the positive EDGE specification. A well-formed maximal projection cannot bear any positive EDGE specifications corresponding to uninterpreted prefixes.

The phrasal affix analysis yields rather untraditional structures (see Figure 3). The proposed treatment provides a straightforward account of contracted forms like "aux" (and the ungrammaticality of uncontracted *"à-les") and of elision and liaison phenomena, all of which call into question the more traditional analysis of these elements as syntactic words.

4.2 Further issues

The notion of phrasal affix is useful for many languages besides French. Candidate phenomena include possessive "-'s" in English, and perhaps the "a"/"an" alternation, case suffixes in Japanese, Korean, and Turkish, and consonant mutation effects in Celtic. In general, phrasal affixes are the remnants of syntactic functors (determiners, prepositions, complementizers) that have lost their morphosyntactic independence, and are now bound morphophonologically to the first (or last) word in a phrase.

All of the phenomena mentioned here involve elements that occupy the frontier between syntax and morphology. In many of these cases, the proposed affixal status of a given element may be in conflict with its orthographic status in the written language. For instance, this is true of the non-elided forms of French "le," "à" and "de" (and their contractions): formally they are analyzed as prefixes but in French text they are written as independent words. At the moment, in our grammar, the lexical rule responsible for "aux" prefixation in (ii), for example, produces a single word `aux_anciens` but ideally we would prefer for the output of the rule to be two (orthographic) words. The same remark applies to pronominal clitics, treated as verbal affixes—throughout the Romance languages, in fact (Miller and Monachesi, in press). The problem of syntax-orthography mismatch recurs in many languages, and calls for a general, multilingual solution.

5 Conclusion

Based on the preliminary results of our monolingual French grammar implementation project, we hope to have identified some issues of multilingual relevance, including some ideas for lexical rule management, and a demonstration of an HPSG

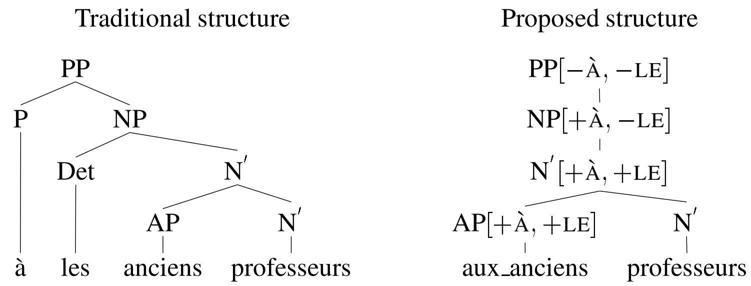


Figure 3: French phrasal affixes

implementation of phrasal affixes. We have also pointed out a few of the problematic aspects of our grammar, since related problems may exist in grammars for other languages, and a multilingual perspective may yield general solutions. It is our hope that the insights gained from our ongoing efforts, and from the discussion generated by this workshop, will be of benefit to similar projects, particularly those devoted to other Romance languages.

References

Anne Abeillé and Danièle Godard. 1996. La complémentation des auxiliaires français. *Langages*, 122:32–61.

Anne Abeillé and Danièle Godard. 1997. Les causatives en français: un cas de compétition syntaxique. *Langue Française*, 115:62–74.

Anne Abeillé and Danièle Godard. 2002. The syntactic structures of French auxiliaries. *Language*, 78:404–452.

Anne Abeillé and Danièle Godard. (in press). Les prédicats complexes dans les langues romanes. In Godard (in press).

Anne Abeillé, Danièle Godard, and Ivan A. Sag. 1998. Two kinds of composition in French complex predicates. In Erhard Hinrichs, Andreas Kathol, and Tsuneko Nakazawa, editors, *Complex Predicates in Nonderivational Syntax*, volume 30 of *Syntax and Semantics*, pages 1–41. Academic Press, New York.

Gosse Bouma, Rob Malouf, and Ivan A. Sag. 2001. Satisfying constraints on extraction and adjunction. *Natural Language and Linguistic Theory*, 19:1–65.

Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Publications, Stanford, CA.

Jonathan Ginzburg and Ivan A. Sag. 2001. *Interrogative Investigations: The Form, Meaning and Use of English Interrogatives*. CSLI Publications, Stanford, CA.

Danièle Godard, editor. (in press). *Les langues romanes. Problèmes de la phrase simple*. CNRS Editions, Paris.

Erhard Hinrichs and Tsuneko Nakazawa. 1994. Linearizing AUXs in German verbal complexes. In John Nerbonne, Klaus Netter, and Carl Pollard, editors, *German in Head-Driven Phrase Structure Grammar*, volume 46 of *CSLI Lecture Notes*, pages 11–37. CSLI Publications, Stanford, CA.

Valia Kordoni, editor. 1999. *Tübingen Studies in Head-Driven Phrase Structure Grammar*. Number 132 in Arbeitspapiere des Sonderforschungsbereichs 340. Universität Tübingen, Tübingen.

Detmar Meurers. 2000. Lexical generalizations in the syntax of German non-finite constructions. Technical Report 145, SFB 340, Universität Tübingen.

Philip Miller and Paola Monachesi. (in press). Les pronoms clitiques dans les langues romanes. In Godard (in press).

Philip H. Miller and Ivan A. Sag. 1997. French clitic movement without clitics or movement. *Natural Language and Linguistic Theory*, 15:573–639.

Philip H. Miller. 1992. *Clitics and Constituents in Phrase Structure Grammar*. Garland, New York.

Jesse Tseng, editor. 2000. *Aspekte eines HPSG-Fragments des Deutschen*. Number 156 in Arbeitspapiere des Sonderforschungsbereichs 340. Universität Tübingen, Tübingen.

Jesse Tseng. 2003a. Edge features and French liaison. In Frank van Eynde, Lars Hellan, and Dorothee Beermann, editors, *Proceedings of the 9th International HPSG Conference*. CSLI Publications, Stanford, CA.

Jesse Tseng. 2003b. Un traitement lexical des affixes syntagmatiques du français. In Bernard Fradin, Georgette Dal, Françoise Kerleroux, Nabil Hathout, Marc Plénat, and Michel Roché, editors, *Les Unités morphologiques / Morphological Units*. SILEX, Villeneuve d'Ascq.

Gertjan van Noord and Gosse Bouma. 1996. Dutch verb clustering without verb clusters. In P. Blackburn and M. de Rijke, editors, *Specifying Syntactic Structures*, pages 1–31. CSLI Publications, Stanford, CA.