

# Lexicalising Word Order Constraints for Implemented Linearisation Grammar

Yo Sato

Department of Computer Science  
King's College London  
yo.sato@kcl.ac.uk

## Abstract

This paper presents a way in which a lexicalised HPSG grammar can handle word order constraints in a computational parsing system, without invoking an additional layer of representation for word order, such as Reape's Word Order Domain. The key proposal is to incorporate into *lexical heads* the WOC (Word Order Constraints) feature, which is used to constrain the word order of its projection. We also overview our parsing algorithm.

## 1 Introduction

It is a while since the linearisation technique was introduced into HPSG by Reape (1993; 1994) as a way to overcome the inadequacy of the conventional phrase structure rule based grammars in handling 'freer' word order of languages such as German and Japanese. In parallel in computational linguistics, it has long been proposed that more flexible parsing techniques may be required to adequately handle such languages, but hitherto a practical system using linearisation has eluded large-scale implementation. There are at least two obstacles: its higher computational cost accompanied with non-CFG algorithms it requires, and the difficulty to state word order information succinctly in a grammar that works well with a non-CFG parsing engine.

In a recent development, the 'cost' issue has been tackled by Daniels and Meurers (2004), who propose to narrow down on search space while using a non-CFG algorithm. The underlying principle is to give priority to the full generative capacity, let the parser overgenerate at default but restrict generation for efficiency thereafter. While sharing this principle, I will attempt to further

streamline the computation of linearisation, focusing mainly on the issue of grammar formalism. Specifically, I would like to show that the lexicalisation of word order constraints is possible with some conservative modifications to the standard HPSG (Pollard and Sag, 1987; Pollard and Sag, 1994). This will have the benefit of making the representation of linearisation grammar simpler and more parsing friendly than Reape's influential Word Order Domain theory.

In what follows, after justifying the need for non-CFG parsing and reviewing Reape's theory, I will propose to introduce into HPSG the Word Order Constraint (WOC) feature for *lexical heads*. I will then describe the parsing algorithm that refers to this feature to constrain the search for efficiency.

### 1.1 Limitation of CFG Parsing

One of the main obstacles for CFG parsing is the discontinuity in natural languages caused by 'interleaving' of elements from different phrases (Shieber, 1985). Although there are well-known syntactic techniques to enhance CFG as in GPSG (Gazdar et al., 1985), there remain constructions that show 'genuine' discontinuity of the kind that cannot be properly dealt with by CFG.

Such 'difficult' discontinuity typically occurs when it is combined with scrambling – another symptomatic phenomenon of free word order languages – of a verb's complements. The following is an example from German, where scrambling and discontinuity co-occur in what is called 'incoherent' object control verb construction.

- (1) Ich glaube, dass der Fritz dem Frank  
I believe Comp Fritz(Nom) Frank(Dat)

das Buch zu lesen erlaubt.  
the book(Acc) to read allow

'I think that Fritz allows Frank to read the book'

- (1') Ich glaube, dass der Fritz [das Buch] dem Frank [zu lesen] erlaubt  
 Ich glaube, dass dem Frank [das Buch] der Fritz [zu lesen] erlaubt  
 Ich glaube, dass [das Buch] dem Frank der Fritz [zu lesen] erlaubt  
 ...

Here (1) is in the ‘canonical’ word order while the examples in (1’) are its scrambled variants. In the traditional ‘bi-clausal’ analysis according to which the object control verb subcategorises for a *zu*-infinitival VP complement as well as nominal complements, this embedded VP, *das Buch zu lesen*, becomes discontinuous in the latter examples (in square brackets).

One CFG response is to use ‘mono-clausal’ analysis or *argument composition* (Hinrichs and Nakazawa, 1990), according to which the higher verb and lower verb (in the above example *erlauben* and *zu lesen*) are combined to form a single *verbal complex*, which in turn subcategorises for nominal complements (*das Buch, der Fritz* and *dem Frank*). Under this treatment both the verbal complex and the sequence of complements are rendered continuous, rendering all the above examples CFG-parseable.

However, this does not quite save the CFG parseability, in the face of the fact that you could extrapose the lower V + NP, as in the following.

- (2) Ich glaube, dass der Fritz dem Frank [erlaubt], das Buch [zu lesen].

Now we have a discontinuity of ‘verbal complex’ instead of complements (the now discontinuous verbal complex is marked with square brackets). Thus either way, some discontinuity is inevitable.

Such discontinuity is by no means a marginal phenomenon limited to German. Parallel phenomena are observed in the object control verbs in Korean and Japanese ((Sato, 2004) for examples). These languages also show a variety of ‘genuine’ discontinuity of other sorts, which do not lend itself to a straightforward CFG parsing (Yatabe, 1996). The CFG-recalcitrant constructions exist in abundance, pointing to an acute need for non-CFG parsing.

## 1.2 Reape’s Word Order Domain

The most influential proposal to accommodate such discontinuity/scrambling in HPSG is Reape’s Word Order Domain, or DOM, a feature that constitutes an additional layer separate from the dominance structure of phrases (Reape, 1993; Reape, 1994). DOM encodes the phonologically realised (‘linearised’) list of signs: the daughter signs of a

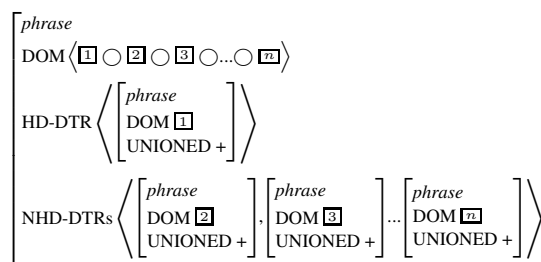


Figure 1: Word Order Domain

phrase in the HD-DTR and NHD-DTRS features are linearly ordered as in Figure 1.

The feature UNIONED in the daughters indicates whether discontinuity amongst their constituents is allowed. Computationally, the positive (+) value of the feature dictates (the DOMs of) the daughters to be *sequence unioned* (represented by the operator ○) into the mother DOM: details apart, this operation essentially merges two lists in a way that allows interleaving of their elements.

In Reape’s theory, LP constraints come from an entirely different source. There is nothing as yet that blocks, for instance, the ungrammatical *zu lesen das Buch* VP sequence. The relevant constraint, i.e. COMPS <ZU-INF-V in German, is stated in the LP component of the theory. Thus with the interaction of the UNIONED feature and LP statements, the grammar rules out the unacceptable sequences while endorsing grammatical ones such as the examples in (1’).

One important aspect of Reape’s theory is that DOM is a list of whole *signs* rather than of any *part* of them such as PHON. This is necessitated by the fact that in order to determine how DOM should be constructed, the daughters’ internal structure need to be referred to, above all, the UNIONED feature. In other words, the internal features of the daughters must be *accessible*.

While this is a powerful system that overcomes the inadequacies of phrase-structure rules, some may feel this is a rather heavy-handed way to solve the problems. Above all, much information is repeated, as all the signs are effectively stated twice, once in the phrase structure and again in DOM. Also, the fact that discontinuity and linear precedence are handled by two distinct mechanisms seems somewhat questionable, as these two factors are computationally closely related. These properties are not entirely attractive features for a computational grammar.

## 2 Lexicalising Word Order Constraints

### 2.1 Overview

Our theoretical goal is, in a nutshell, to achieve what Reape does, namely handling discontinuity and linear precedence, in a simpler, more lexicalist manner. My central proposal consists in incorporating the Word Order Constraint (WOC) feature into the lexical heads, rather than positing an additional tier for linearisation. Some new subfeatures will also be introduced.

The value of the WOC feature is a set of word-order related constraints. It may contain any relational constraint the grammar writer may want with the proviso of its formalisability, but for the current proposal, I include two subfeatures ADJ (adjacency) and LP, both of which, being binary relations, are represented as a set of ordered pairs, the members of which must either be the head itself or its sisters. Figure 2 illustrates what such feature structure looks like with an English verb *provide*, as in *provide him with a book*.

We will discuss the new PHON subfeatures in the next section – for now it would suffice to consider them to constitute the standard PHON list – so let us focus on WOC here. The WOC feature of this verb says, for its projection (VP), three constraints have to be observed. Firstly, the ADJ subfeature says that the indirect object NP has to be in the adjacent position to the verb (*provide yesterday him with a book* is not allowed). Secondly, the first two elements of the LP value encode a head-initial constraint for English VPs, namely that a head verb has to be preceded by its complements. Lastly, the last pair in the same set says the indirect object must precede the with-PP (*provide with a book him* is not allowed). Notice that this specification leaves room for some discontinuity, as there is no ADJ requirement between the indirect NP and with-PP. Hence, *provide him yesterday with a book* is allowed.

The key idea here is that since the complements of a lexical head are available in its COMPS feature, it should be possible to state the relative linear order which holds between the head and a complement, as well as between complements, *inside* the feature structure of the head.

Admittedly word order would naturally be considered to reside in a phrase, string of words. It might be argued, on the ground that a head’s COMPS feature simply consists of the *categories* it selects for in exclusion of the PHON feature, that with this architecture one would inevitably encounter the ‘accessibility’ problem discussed in

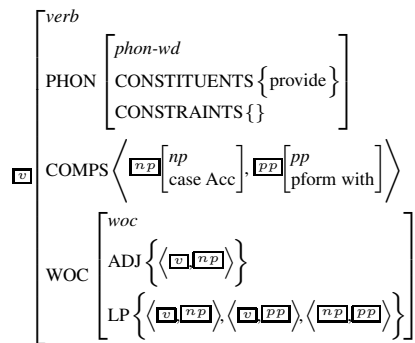


Figure 2: Example of lexical head with WOC feature

Section 1.2: in order to ensure the enforceability of word order constraints, an access must be secured to the values of the internal features including the PHON values. However, this problem can be overcome, as we will see, if due arrangements are in place.

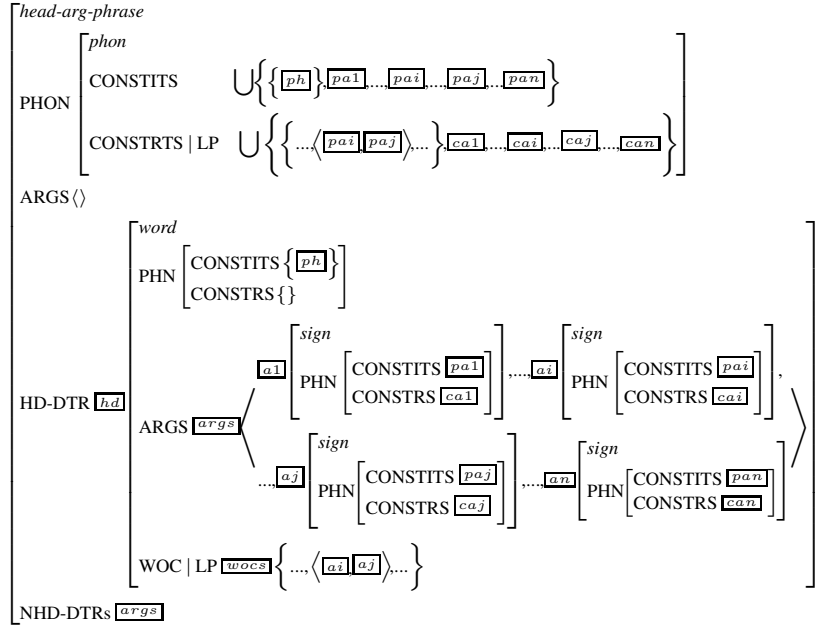
The main benefit of this mechanism is that it paves way to an entirely lexicon-based rule specification, so that, on one hand, duplication of information between lexical specification and phrase structure rules can be reduced and on the other, a wide variety of lexical properties can be flexibly handled. If the word order constraints, which have been regarded as the bastion of rule-based grammars, is shown to be lexically handled, it is one significant step further to a fully lexicalist grammar.

### 2.2 New Head-Argument Schema

What is crucial for this WOC-incorporated grammar is how the required word order constraints stated in WOC are passed on and enforced in its projection. I attempt to formalise this in the form of *Head-Argument Schema*, by modifying *Head-Complement Schema* of Pollard and Sag (1994). There are two key revisions: an enriched PHON feature that contains word order constraints and percolation of these constraints emanating from the WOC feature in the head.

The revised Schema is shown in Figure 3. For simplicity only the LP subfeature is dealt with, since the ADJ subfeature would work exactly the same way. The set notations attached underneath states the restriction on the value of WOC, namely that all the signs that appear in the constraint pairs must be ‘relevant’, i.e. must also appear as daughters (included in ‘DtrSet’, the set of the head daughter and non-head daughters). Naturally, they also cannot be the same signs ( $x \neq y$ ).

Let me discuss some auxiliary modifications



where  $wocs \subseteq \{ \langle x, y \rangle \mid x \neq y, x, y \in DtrSet \}$   
 $DtrSet = \{ [hd] \} \cup [args]$

Figure 3: Head-Argument Schema with WOC feature

first. Firstly, we change the feature name from COMPS to ARGS because we assume a non-configurational flat structure, as is commonly the case with linearisation grammar. Another change I propose is to make ARGS a list of underspecified *signs* instead of SYNSEMs as standardly assumed (Pollard and Sag, 1994). In fact, this is a position taken in an older version of HPSG (Pollard and Sag, 1987) but rejected on the ground of the locality of subcategorisation. The main reason for this reversal is to facilitate the ‘accessibility’ we discussed earlier. As unification and percolation of the PHON information is involved in the Schema, it is much more straightforward to formulate with signs. Though the change may not be quite defensible solely on this ground,<sup>1</sup> there is reason to leave the locality principle as an option for languages of which it holds rather than hardwire it into the Schema, since some authors raise doubt as for the universal applicability of the locality principle e.g. (Meurers, 1999).

Turning to a more substantial modification, our new PHON feature consists of two subfeatures, CONSTITUENTS (or CONSTITS) and CONSTRAINTS (or CONSTRS). The former encodes the set that comprises the phonology of words of which the string consists. Put simply, it is the un-

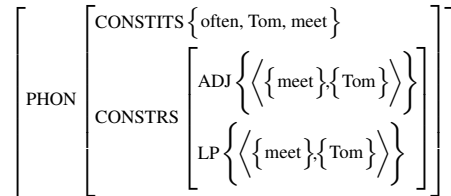
ordered version of the standard PHON list. The CONSTRAINTS feature represents the concatenative constraints applicable to the string. Thus, the PHON feature overall represents the legitimate word order patterns in an underspecified way, i.e. any of the possible string combinations that obey the constraints. Let me illustrate with a VP example, say, consisting of *meet*, *often* and *Tom*, for which we assume that the following word order patterns are acceptable,

$\langle \text{meet, Tom, often} \rangle, \langle \text{often, meet, Tom} \rangle$

but not the followings:

$\langle \text{meet, often, Tom} \rangle, \langle \text{Tom, often, meet} \rangle,$   
 $\langle \text{Tom, meet, often} \rangle, \langle \text{often, Tom, meet} \rangle.$

This situation can be captured by the following feature specification for PHON, which encodes any of the acceptable strings above in an underspecified way.



The key point is that now the computation of word order can be done based on the information inside the PHON feature, though indeed the CONSTR values have to come from outside – the word order crucially depends on SYNSEM-related values of the daughter signs.

<sup>1</sup>Another potential problem is cyclicity, since the sign-valued ARGS feature contains the WOC feature, which could contain the head itself. This has to be fixed for the systems that do not allow cyclicity.

Let us now go back to the Schema in Figure 3 and see how to determine the CONSTR values to enter the PHON feature. This is achieved by looking up the WOC constraints in the head (let's call this Step 1) and pushing the relevant constraints into the PHON feature of its mother, according to the type of constraints (Step 2).

For readability Figure 3 only states explicitly a special case – where one LP constraint holds of two of the arguments – but the reader is asked to interpret  $\overline{a1}$  and  $\overline{a2}$  in the head daughter's WOC|LP to represent any two signs chosen from the 'DTRS' list (including the head,  $\overline{v1}$ ).<sup>2</sup> The structure sharing of  $\overline{a1}$  and  $\overline{a2}$  between WOC|LP and ARGS indicates that the LP constraint applies to these two arguments in this order, i.e.  $\overline{a1} \prec \overline{a2}$ . Thus through unification, it is determined which constraints apply to which pairs of daughter signs inside the head. This corresponds to Step 1.

Now, only for these WOC-applicable daughter signs, the PHON|CONSTITS values are paired up for each constraint (in this case  $\langle \overline{pa1}, \overline{pa2} \rangle$ ) and pushed into the mother's PHON|CONSTRS feature. This corresponds to Step 2.

Notice also that the CONSTRAINTS subfeature is *cumulatively* inherited. All the non-head daughters' CONSTR values ( $\langle \overline{ca1}, \dots, \overline{cam} \rangle$ ) – the word order constraints applicable to each of these daughters – are also passed up, collecting effectively all the CONSTR values of its daughters and descendants. This means the information concerning word order, as tied to particular string pairs, is never lost and passed up all the way through. Thus the WOC constraints can be enforced at any point where both members of the string pair in question are instantiated.

### 2.3 A Worked Example

Let us now go through an example of applying the Schema, again with the German subordinate clause, *das Buch der Fritz dem Frank zu lesen erlaubt* (and other acceptable variants). Our goal is to enforce the ADJ and LP constraints in a flexible enough way, allowing the acceptable sequences such as those we saw in Section 1.2.1. while blocking the constraint-violating instances.

The instantiated Schema is shown in Figure 4. Let us start with a rather deeply embedded level, the embedded verb *zu-lesen*, marked  $\overline{v2}$ , found inside  $\overline{v1}$  (the last and largest NHD-DTR) as its HD-

DTR, which I suppose to be one lexical item for simplicity. This is one of the lexical heads from which the WOC constraints emanate. Find, in this item's WOC, a general LP constraint for zu-Infinitiv VPs, COMPS  $\prec$  V, namely  $\overline{vp3} \prec \overline{v2}$ . Then the PHON|CONSTITS values of these signs are searched for and found in the daughters, namely  $\overline{pnp3}$  and  $\overline{pv2}$ . These values are paired up and passed into the CONSTRS|LP value of its mother VP. Notice also that into this value the NHD-DTRS' CONSTR|LP values, in this case only  $\overline{pnp3}$  ( $\{ \text{das} \} \prec \{ \text{Buch} \}$ ), are also unioned, constituting  $\overline{pvp2}$ : we are here witnessing the cumulative inheritance of constraints explained earlier. Turn attention now to the percolation of ADJ subfeature: no ADJ requirement is found between *das Buch* and *zu-lesen* ( $\overline{v2}$ 's WOC|ADJ is empty), though ADJ is required one node below, between *das* and *Buch* ( $\overline{pnp3}$ 's PHN|CONSTR|ADJ). Thus no new ADJ pair is added to the mother VP's PHON|CONSTR feature.

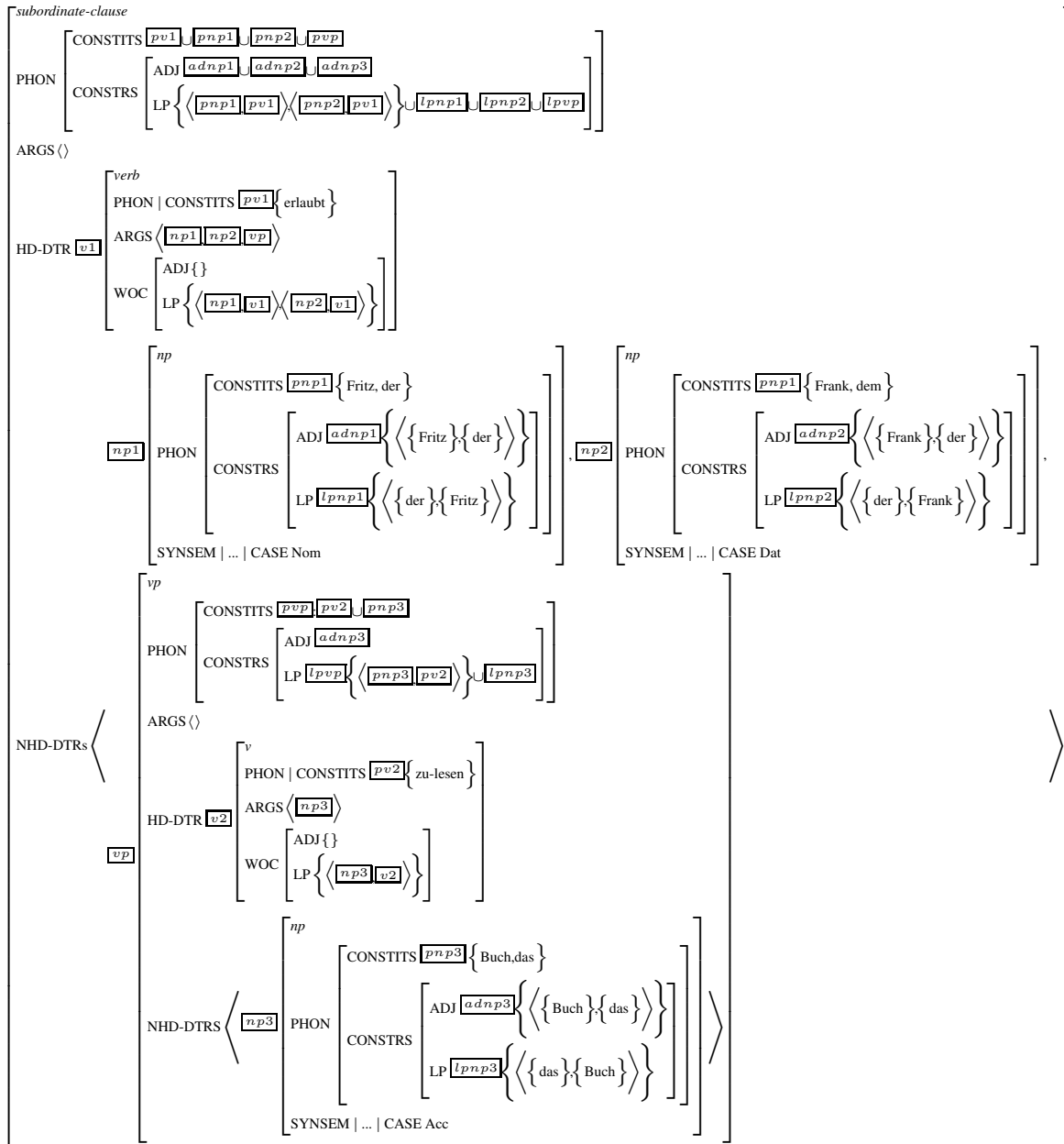
Exactly the same process is repeated for the projection of *erlauben* ( $\overline{v1}$ ), where its WOC again contains only LP requirements. With the PHON|CONSTITS values of the relevant signs found and paired up ( $\{ \text{Fritz, der} \} \prec \{ \text{erlaubt} \}$  and  $\{ \text{Frank, dem} \} \prec \{ \text{erlaubt} \}$ ), they are pushed into its mother's PHON|CONSTRS|LP value, which is also unioned with the PHON|CONSTRS values of the NHD-DTRS. Notice this time that there is no LP requirement between the zu-Infinitiv VP, *das Buch zu-lesen*, and the higher verb, *erlaubt*. This is intended to allow for extraposition.<sup>3</sup>

The eventual effect of the cumulative constraint inheritance can be more clearly seen in the sub-AVM underneath, which shows the PHON part of the whole feature structure with its values instantiated. After a succession of applications of the Head-Argument Schema, we now have a pool of WOCs sufficient to block unwanted word order patterns while endorsing legitimate ones. The representation of the PHON feature being underspecified, it corresponds to any of the appropriately constrained order patterns. *der Fritz dem Frank zu lesen das Buch erlaubt* would be ruled out by the violation of the last LP constraint, *der Fritz erlaubt dem Frank das Buch zu lesen* by the second, and so on.

The reader might be led to think, because of

<sup>2</sup>For the generality of the number of ARGS elements, which should be taken to be any number including zero, the recursive definition as detailed in (Richter and Sailer, 1995) can be adopted.

<sup>3</sup>The lack of this LP requirement also entails some marginally acceptable instances, such as *der Fritz dem Frank das Buch erlaubt zu lesen*, considered ungrammatical by many. These instances can be blocked, however, by introducing more complex WOCs. See Sato (forthcoming a).



Instantiated PHON part of the above:

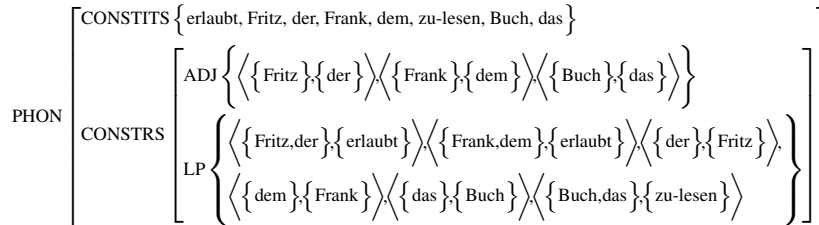


Figure 4: An application of Head-Argument Schema

the monotonic inheritance of constraints, that the WOC compliance cannot be checked until the stage of final projection. While this is *generally* true for freer word order languages considering various scenarios such as bottom-up generation, one can conduct the WOC check *immediately* after the instantiation of relevant categories in parsing, the fact we can exploit in our implementation, as we will now see.

### 3 Constrained Free Word Order Parsing

#### 3.1 Algorithm

In this section our parsing algorithm that works with the lexicalised linearisation grammar outlined above is briefly overviewed.<sup>4</sup> It expands on two existing ideas: bitmasks for non-CFG parsing and dynamic constraint application.

Bitmasks are used to indicate the positions of a parsed words, *wherever* they have been found. Reape (1991) presents a non-CFG tabular parsing algorithm using them, for ‘permutation complete’ language, which accepts all the permutations and discontinuous realisations of words. To take for an example a simple English NP that comprises *the*, *thick* and *book*, this parser accepts not only their 3! permutations but discontinuous realisations thereof in a longer string, such as [book, -, the, -, thick] (‘-’ indicates the positions of constituents from other phrases).

Clearly, the problem here is overgeneration and (in)efficiency. In the current form the worst-case complexity will be exponential ( $O(n! \cdot 2^n)$ ,  $n$  = length of string). In response, Daniels and Meurers (2004) propose to restrict search space during the parse with two additional bitmasks, *positive* and *negative masks*, which encode the bits that must be and must not be occupied, respectively, based on what has been found thus far and the relevant word order constraints. For example, given the constraints that Det precedes Nom and Det must be adjacent to Nom and supposing the parser has found Det in the third position of a five word string like above, the negative mask [ x, x, the, -, -] is created, where x indicates the position that cannot be occupied by Nom, as well as the positive mask [ \*, \*, \*, -, -], where \* indicates the positions that must be occupied by Nom. Thus, you can stop the parser from searching the positions the categories yet to be found cannot occupy, or force it to search only the positions they have to occupy.

<sup>4</sup>For full details see Sato (forthcoming b).

A remaining important job is to how to state the constraints themselves in a grammar that works with this architecture, and Daniels and Meurers’ answer is a rather traditional one: stating them in phrase structure rules as LP attachments. They modify HPSG rather extensively in a way similar to GPSG, in what they call ‘Generalised ID/LP Grammar’. However, as we have been arguing, this is not an inevitable move. It is possible to keep the general contour of the standard HPSG largely intact.

The way our parser interacts with the grammar is fundamentally different. We take full advantage of the information that now resides in lexical heads. Firstly, rules are dynamically generated from the subcategorisation information (ARGS feature) in the head. Secondly, the constraints are picked up from the WOC feature when lexical heads are encountered and carried in edges, eliminating the need for positive/negative masks. When an active edge is about to embrace the next category, these constraints are checked and enforced, limiting the search space thereby.

After the lexicon lookup, the parser generates rules from the found lexical head and forms lexical edges. It is also at this stage that the WOC is picked up and pushed into the edge, along with the rule generated:

(Mum → Hd-Dtr • Nhd<sub>1</sub> Nhd<sub>2</sub>...Nhd<sub>n</sub>; WOCs)

where WOCs is the set of ADJ and LP constraints picked up, if any. This edge now tries to find the rest – non-head daughters. The following is the representation of an edge when the parsing proceeds to the stage where some non-head daughter, in this representation Dtr<sub>i</sub>, has been parsed, and Dtr<sub>j</sub> is to be searched for.

(Mum → Dtr<sub>1</sub> Dtr<sub>2</sub>...Dtr<sub>j</sub> • Dtr<sub>j</sub>...Dtr<sub>n</sub>; WOCs)

When Dtr<sub>j</sub> is found, the parser does *not* immediately move the dot. At this point the WOC compliance check with the relevant WOC constraint – the one(s) involving Dtr<sub>i</sub> and Dtr<sub>j</sub> – is conducted on these two daughters. The compliance check is a simple list operation. It picks the bitmasks of the two daughters in question and checks whether the occupied positions of one daughter precede/are adjacent to those of the other.

The failure of this check would prevent the dot move from taking place. Thus, edges that violate the word order constraints would not be created, thereby preventing wasteful search. This is the same feature as Daniels and Meurers’, and therefore the efficiency in terms of the number of edges is identical. The main difference is that we use

the information inside the feature structure without having media like positive/negative masks.

### 3.2 Implementation

I have implemented the algorithm in Prolog and coded the HPSG feature structure in the way described using ProFIT (Erbach, 1995). It is a head-corner, bottom-up chart parser, roughly based on Gazdar and Mellish (1989). The main modification consists of introducing bitmasks and the word order checking procedure described above. I created small grammars for Japanese and German and put them to the parser, to confirm that linearisation-heavy constructions such as object control construction can be successfully parsed, with the WOC constraints enforced.

### 4 Future Tasks

What we have seen is an outline of my initial proposal and there are numerous tasks yet to be tackled. First of all, now that the constraints are written in individual lexical items, we are in need of appropriate typing in terms of word order constraints, in order to be able to state succinctly general constraints such as the head-final/initial constraint. In other words, it is crucial to devise an appropriate type hierarchy.

Another potential problem concerns the generality of our theoretical framework. I have focused on the Head-Argument structure in this paper, but if the present theory were to be of general use, non-argument constructions, such as the Head-Modifier structure, must be accounted for. Also, the cases where the head of a phrase is itself a phrase may pose a challenge, if such a phrasal head were to determine the word order of its projection. Since it is desirable for computational transparency not to use emergent constraints, I will attempt to get all the word order constraints ultimately propagated and monotonically inherited from the lexical level. Though some word order constraints may turn out to have to be written into the phrasal head directly, I am confident that the majority, if not all, of the constraints can be stated in the lexicon. These issues are tackled in a separate paper (Sato, forthcoming a).

In terms of efficiency, more study has to be required to identify the exact complexity of my algorithm. Also, with a view to using it for a practical system, an evaluation of the efficiency on the actual machine will be crucial.

### References

- M. Daniels and D. Meurers. 2004. GIDL: A grammar format for linearization-based HPSG. In *Proceedings of the HPSG04 Conference*.
- G. Erbach. 1995. ProFIT: Prolog with features, inheritance and templates. *Proceedings of the Seventh Conference of the European Association for Computational Linguistics*.
- G. Gazdar and C. Mellish. 1989. *Natural Language Processing in Prolog*. Addison Wesley.
- G. Gazdar, E. Klein, G. Pullum, and I. Sag. 1985. *Generalized Phrase Structure Grammar*. Harvard UP.
- E. Hinrichs and T. Nakazawa. 1990. Subcategorization and VP structure in German. In S. Hughes et al., editor, *Proceedings of the Third Symposium on Germanic Linguistics*.
- D. Meurers. 1999. Raising Spirits (and assigning them case). *Groninger Arbeiten zur Germanistischen Linguistik, Groningen Univ.*
- C. Pollard and I. Sag. 1987. *Information-Based Syntax and Semantics*. CSLI.
- C. Pollard and I. Sag. 1994. *Head-Driven Phrase Structure Grammar*. CSLI.
- M. Reape. 1991. Parsing bounded discontinuous constituents: Generalisation of some common algorithms. *DIANA Report, Edinburgh Univ.*
- M. Reape. 1993. *A Formal Theory of Word Order*. Ph.D. thesis, Edinburgh University.
- M. Reape. 1994. Domain union and word order variation in German. In J. Nerbonne et al., editor, *German in Head-Driven Phrase Structure Grammar*.
- F. Richter and M. Sailer. 1995. Remarks on linearization. Magisterarbeit, Tübingen Univ.
- Y. Sato. 2004. Discontinuous constituency and non-CFG parsing. <http://www.dcs.kcl.ac.uk/pg/satoyo>.
- Y. Sato. forthcoming a. Two alternatives for lexicalist linearisation grammar: Locality Principle revisited.
- Y. Sato. forthcoming b. Constrained free word order parsing for lexicalist grammar.
- S. Shieber. 1985. Evidence against the context freeness of natural languages. *Linguistics and Philosophy*, 8(3):333–43.
- S. Yatabe. 1996. Long-distance scrambling via partial compaction. In M. Koizumi et al., editor, *Formal Approaches to Japanese Linguistics 2*. MIT Press, Cambridge, Mass.