

# DIRECT PARSING WITH METARULES

Wilhelm WEISWEBER, Susanne PREUB  
Technical University of Berlin  
Institute for Software and Theoretical Computer Science  
Project KIT-FAST, Sekr. 1/R 5-12  
Franklinstr. 28/29, D-1000 Berlin 10  
E-mail: ww@kit.cs.tu-berlin.de and preuss@kit.cs.tu-berlin.de

## Abstract

In this paper we argue for the direct application of metarules in the parsing process and introduce a slight restriction on metarules. This restriction relies on theoretical results about the termination of term-rewrite systems and does not reduce the expressive power of metarules as much as previous restrictions. We prove the termination for a set of metarules used in our German grammar and show how metarules can be integrated into the parser.

## 1 Introduction

The metarules within the theory of Generalized Phrase Structure Grammar (GPSG) are a very interesting device to express generalizations on a basic set of immediate dominance (ID) rules. A short introduction to the theory of metarules is given in the following section. Metarules are used to generate an object grammar from a set of basic ID rules. One of the first discussions about the application time of metarules within the GPSG theory, which has strongly influenced the succeeding research in this field, has been published by [Thompson 82]. In his article Thompson argued for applying the metarules all at once in a preprocessing step (compile-time application). However, our parser applies the metarules during the parsing process (run-time application or direct application). A discussion why we prefer the direct application is given in section 3. No matter when the metarules are applied to the basic set of ID rules, we have to care for the termination of the recursive application of metarules. [Thompson 82] made a proposal to guarantee termination by the definition of a Finite Closure (FC). This approach restricts the application of metarules such that one metarule can only be applied once in the derivation of an ID rule and prevents it from recursive application. This restriction has been taken over by the authors of [Gazdar et al. 85] and they gave a further restriction. They restricted the application of metarules to lexical ID rules, even though they knew that this restriction may prove to be incompatible with the descriptive power needed for natural language grammars (p. 59). But we think that there is no need to restrict the application of metarules only to lexical ID rules, even if there are proposals to eliminate the use of metarules and to use lexical rules like in L1-G and HPSG (see [Jacobson 87]). But to do so with GPSG would involve crucial changes to the theory, and therefore we preserved the metarule component in our machine translation (MT) system and tried to find an adequate criterion for the termination of metarules. In our approach the grammar writer is free to decide whether a metarule is to be applied to lexical ID rules or to another type of rule.

The Finite Closure (FC) is too restrictive, because in some cases (see the examples of [Uszkoreit 87] and [Gazdar et al. 85] in the sections 3 and 4, respectively) recursive application is needed. In section 4 we present an alternative constraint on the basis of results

in the field of term-rewriting. This constraint is less restrictive than the FC. It allows for the definition of recursive metarules which may be applied freely and guarantees the termination. In section 5 the metarules of the German GPSG grammar used in our MT system are outlined and, with the help of the constraint in section 4, we show that their application is terminating. In section 6 we give an outline of how to modify the parsing process in a way that metarules can be applied directly rather than at compile-time. In the last section we draw some conclusions for future work in this field.

## 2 Metarules in GPSG

Metarules are one of the most criticized devices of the GPSG formalism. GPSG is a grammar formalism that states most of its generalizations on the level of local trees. Metarules were introduced to capture generalizations on the set of ID rules. An ID rule states the dominance relation between the mother category and a multiset of daughter categories in a local tree without fixing the linear precedence relation of the daughters. ID rules have the following format:

$$C_0 \rightarrow C_1, C_2, \dots, C_n$$

Metarules define a relation on ID rules. They have the following format:

$$\text{'input ID rule scheme'} \Rightarrow \text{'output ID rule scheme'}$$

and can be read as: If the set of ID rules contains an ID rule which is matched by 'input ID rule scheme', then it also contains an ID rule that matches 'output ID rule scheme', where the feature specifications of the input ID rule are carried over to the output ID rule if not specified otherwise by the metarule. For example the metarule  $VP[+PAS] \rightarrow W, NP[acc] \Rightarrow VP[+PAS] \rightarrow W, (PP[by])$  states the connection between active and passive, where  $W$  is a variable ranging over a (possibly empty) multiset of categories. The major point of criticism against metarules is that they increase the generative power of GPSG in an undesirable way when they are recursively applicable, because this may lead to an infinite set of ID rules. The resulting grammar need not be context free. In order to remedy the situation, suggestions of varying radicality were made. The proposal of [Thompson 82] and [Gazdar et al. 85], which tries to maintain metarules, was simply to apply a metarule at most once in the generation of an ID rule. This stipulation is somewhat strange, because it allows for recursive metarules and just prevents them from being applied recursively.

[Kilbury 86] suggested to eliminate metarules by using category cooccurrence restrictions.

The most radical proposal was to dispense with metarules. But our aim was to stay within the framework of GPSG, and it would be a loss to dispense with metarules, because GPSG formulates for example valency of verbs and other constituents on the level of ID rules and metarules are the means to capture generalisations on that level.

For this reason we formalize the properties of metarules that terminate recursive application and state

them as a condition that a set of metarules must fulfil. Metarules can then be applied freely.

### 3 Application time

There are two possibilities for the time to apply the metarules. The first is to compile the basic set of ID rules (compile-time application) in a preprocessing step. Thompson calls it all-at-once approach. The other possibility is to apply the metarules during the parsing process (run-time application or direct application), which Thompson calls as-needed approach. Thompson argued for the compile-time application because the direct application of metarules has the following disadvantages (see [Thompson 82]: p.5):

- (1) If a metarule can be applied to an ID rule during the parsing process, the metarule has to be applied again when the same ID rule is involved in the same or a subsequent parse.
- (2) To store the structures generated by ID rules which are the result of the application of a metarule is just another instance of the compile-time application.
- (3) Derivations of ID rules of length greater than one, i.e. ID rules which are the results of applying more than one metarule to one basic ID rule, will rapidly expand the search space.

In order to look a little bit closer to Thompson's arguments and to stay on his line, we presuppose that a kind of top-down parsing method is used and there are  $n$  basic ID rules and  $m$  ID rules, generated by the application of the metarules.

When looking to argument (1), we see that it is an argument for the run-time approach. If the metarules are applied at compile-time a huge set of ID rules is compiled from the basic set. For example if we would apply the metarules of our MT system (see section 5) to our basic set of 80 ID rules at compile-time, we would get about 240 ID rules in the object grammar. Let us assume that some category  $C$  has to be expanded and there are  $i$  ID rules in our grammar with mother category  $C$ . In the compile-time approach the parser would have to check  $(n+m)/n \cdot i$  ID rules on average, whereas in the run-time approach  $i$  ID rules and  $(n+m)/n$  metarules  $((n+m)/n \cdot i$  rules) have to be checked for application to these ID rules. In the normal case that are less than in the compile-time approach.

Argument (2) is indeed an argument for the run-time approach. Let us again consider the above example and each of the  $i$  ID rules has  $d$  daughters on average. If the category  $C$  is expanded according to all ID rules, in the worst case  $(n+m)/n \cdot d \cdot i$  (partial) structures have to be stored on average in the compile-time approach. These structures are very similar, because in general the metarules modify the ID rules slightly. The run-time approach can make use of this fact and stores only approximately  $d \cdot i$  (partial) structures and additionally  $(n+m)/n \cdot i$  structures after the application of the metarules. That makes  $((n+m)/n + d) \cdot i$  structures to be stored in the run-time approach. The common parts of the ID rules generated by metarules need not to be stored, that are  $(n+m)/n \cdot (d-1) \cdot i - d \cdot i$  partial structures less. For example if  $n = 80$ ,  $m = 160$ ,  $d = 3$  and  $i = 10$  then this would mean that on average 30 partial structures less have to be stored for the corresponding constituent.

Concerning argument (3), [Barton et al. 87]: p.226 showed that the computation of the Finite Closure (FC) of a GPSG  $G$  with  $x$  ID rules and  $y$  metarules can

increase the number of the ID rules worse than exponentially, namely from  $x$  to more than  $x^{2y}$ , i.e. there is no difference between the compile-time and run-time approach.

In order to sum up this discussion, we can say that there is no difference in complexity between the compile-time and the run-time approach with respect to the arguments in [Thompson 82]. The direct approach is even preferable to the compile time approach when looking at the arguments (1) and (2), which are indeed arguments for direct application of metarules.

There is another argument for direct application of metarules. The FC in [Thompson 82] states that every metarule can apply at most once in the derivation of any given object grammar rule from one basic rule. An example for the recursive application of a metarule has been proposed in [Uszkoreit 87]: p.145 in his German grammar. It makes the adverbial phrase (AdvP) a sister node of the verb and its arguments:

$$V^2[-AUX] \rightarrow V, W \Rightarrow V^2[-AUX] \rightarrow V, W, AdvP$$

This metarule is to solve the problem that adverbial phrases can be interspersed freely among the arguments of the verb and the number of AdvPs in a verb phrase is not limited (but it is finite) and the metarule has to be applied recursively. This fact would rule out the definition of this metarule with respect to the definition of the FC which has been adopted also in [Gazdar et al. 85] to avoid the production of an infinite number of ID rules. Uszkoreit tried to circumvent the problem in redefining the above metarule such that it fits the requirements of the FC. It employs the Kleene star operator:

$$V^2[-AUX] \rightarrow V, W \Rightarrow V^2[-AUX] \rightarrow V, W, AdvP^*$$

This change of the metarule is not necessary if the metarules are applied directly during the parsing process and the above metarule without the Kleene star can be applied freely, because the termination is determined by the finiteness of the input string.

### 4 Termination

No matter whether the metarules are applied directly or at compile-time, we have to care for the termination. We think that the restriction on the application of metarules imposed by the Finite Closure (FC) is too strong. That may have crucial consequences for some metarules. Look for example at the *Complement Omission Metarule* from the English grammar in [Gazdar et al. 85], which is  $[+N, BAR 1] \rightarrow N^0, W \Rightarrow [+N, BAR 1] \rightarrow N^0$ . This metarule deletes optional complements of nouns. For example the noun *gift* may have two optional prepositional phrases as complements, i.e.  $N^1 \rightarrow N^0, (PP[of]), (PP[to])$ . The prepositions *of* and *to* are fixed but either of the PPs or both may be left out: *the gift, the gift of John, the gift to Paul, the gift of John to Paul*. The above metarule only allows for *the gift and the gift of John to Paul*, because it deletes all complements of the noun. The correct metarule would be:

$$[+N, BAR 1] \rightarrow N^0, X, W \Rightarrow [+N, BAR 1] \rightarrow N^0, W$$

[Gazdar et al. 85] have not been able to define this metarule, because it has to be applied recursively and the FC does not allow recursive application. For this reason we decided to define another constraint which is not so restrictive.

As it has turned out, most of the metarules which have been defined within the fragment of English in [Gazdar et al. 85] or of German (see the section below for a more detailed discussion) are or can be defined in a

way that guarantees the termination of the recursive application of metarules. In order to prove the termination, some research results within the field of term-rewriting can be applied (see [Dershowitz 82 and 85] for general results and [Weisweber 89], [Weisweber/ Hauenschild 90] and [Weisweber 92] for an application to mappings within machine translation). ID rules can be viewed as terms and metarules can be viewed as term-rewrite rules, because they derive one ID rule from another. A set of term-rewrite rules terminates if an ordering '>' on the terms of the left-hand and right-hand sides (lhs and rhs, respectively) of the rewrite rules can be defined. This may be a quantitative ordering, e.g. a category occurring on the lhs of a metarule is deleted on its rhs, or a qualitative ordering, e.g. an operator precedence. We think that a mixture of both types of orderings is needed to prove the termination of sets of metarules. If a qualitative and a quantitative ordering are merged, the resulting ordering guarantees termination (see [Dershowitz 82 and 85]). The operator precedence that is used in our MT system is in fact a precedence ordering on feature values occurring at the categories of the lhs and rhs of the metarules.

#### Termination condition for metarule application

For every metarule  $lhs \Rightarrow rhs, lhs > rhs \Leftrightarrow$

- (i) a daughter category occurring on the lhs is deleted on the rhs **and/or**
- (ii) an operator precedence  $>_{op}$  on feature values occurring at the categories of the lhs and rhs can be defined, which is not contradictory for the whole set of metarules **and**
- (iii) every variable for (multisets of) categories occurring on the rhs occurs on the lhs.

Metarules have to fulfil the conditions (i) and (iii) or (ii) and (iii). The condition (i) is a quantitative ordering and the termination of metarules which fulfil (i) is obvious, because everytime such a metarule is applied one category is deleted and the number of daughter categories in an ID rule is finite.

The condition (ii) is a qualitative ordering. The termination of metarules which fulfil (ii) is not as obvious as in (i). It means that a feature value of a category has to be changed and there must not be another metarule, which reverses the change of this feature value. It allows for not having to delete categories, adding categories or adding values to a list, which may be a feature value, on the rhs of a metarule, if a feature value is changed on another category. This is the reason why we decided to impose control on the definition of metarules and not to move away from such devices as recent research in computational linguistics does. If a metarule fulfils condition (ii) it cannot be applied for a second time in a derivation of an ID rule, because once a feature value has been changed it will never be reversed and the metarule will not be applicable again. This part of the termination condition simulates the termination condition of the FC.

The condition (iii) prohibits the introduction or doubling of variables for (multisets of) categories on the rhs.

Thus the termination of a certain set of metarules can be guaranteed, iff for all metarules either the metarule deletes a category occurring on its lhs and/or a non-contradicting precedence on operators (feature values of the categories occurring in the metarule) can be defined and all variables occurring on the rhs occur on the lhs.

The application of the termination condition is demon-

strated with the metarules for German in the next section.

## 5 Metarules for German

Our GPSG syntax of German is based on the grammar developed in [Uzskoreit 87]. We assume a flat structure for the verb and its complements including the subject. Subcategorization of verbs is stated in ID rules of the following form:

$$V^3 \rightarrow V^0[\text{SUBCAT } \alpha_n], C_1, \dots, C_n$$

$V^3$  is a sentential category and  $\text{SUBCAT } \alpha_n$  means the fixed value for the subcategorization feature in the ID rule with  $n$  arguments, but for every subcategorization there is a separate rule. The subject of main verbs is included in the rhs of the rule. Unlike Uzskoreit's approach we do not add the subject to the complements of a verb phrase via metarule application but reduce a sentential category to a verb phrase and delete the subject. The following *Subject Deletion Metarule* fulfils the termination conditions (i) and (iii), because it deletes the category  $\text{DP}[\text{nom}]$ .<sup>1</sup>

#### Subject Deletion Metarule:

$$\begin{aligned} V^3[-\text{AUX}] &\rightarrow V^0, W, \text{DP}[\text{nom}] \\ &\Rightarrow \\ V^2[-\text{AUX}] &\rightarrow V^0, W \end{aligned}$$

Additionally, the operator precedence 'BAR 3'  $>_{op}$  'BAR 2' has to be defined, because the feature BAR at the mother category is changed on the rhs. This additional definition is needed in order to get a non-contradicting set of operator precedence definitions out of the whole set of metarules in the grammar.

The *Slash Termination Metarule* is responsible for ending (or from the bottom-up view, for the introduction of) a long distance relationship that is handled in GPSG via the category-valued feature SLASH. Unlike [Gazdar et al. 85] we do not have a trace. Traces cause problems in flat structures without fixed word order, because there are multiple analyses that are only different with respect to the position of the trace.

#### Slash Termination Metarules:

$$\begin{aligned} V^3[-\text{AUX}] &\rightarrow V^0, W, X^2 \\ &\Rightarrow \\ V^3[-\text{AUX}, \text{SLASH } X^2] &\rightarrow V^0, W \\ V^3[-\text{AUX}] &\rightarrow V^0, W, V^3 \\ &\Rightarrow \\ V^3[-\text{AUX}, \text{SLASH } V^3] &\rightarrow V^0, W. \end{aligned}$$

Here the termination conditions (i) and (iii) are also fulfilled, because a category of the rhs of the ID rule is deleted. The operator precedence definitions are 'SLASH  $\rightarrow^2$ '  $>_{op}$  'SLASH  $X^2$ ' and 'SLASH  $\rightarrow^1$ '  $>_{op}$  'SLASH  $V^3$ ', respectively.

The *Extraposition Metarule* handles complement sentences and infinitive constructions that we treat as dislocated when they appear in the final position of a sentence. Another category-valued feature, SLASH1, is

<sup>1</sup> The category DP is a determiner phrase according to the X-Bar-Schema in the Government and Binding Theory.

<sup>2</sup> 'F  $\rightarrow$ ' means that the feature F has the value ' $\rightarrow$ ' (see [Busemann/ Hauenschild 88] and [Busemann/ Hauenschild 89]). This is equivalent to the notation  $-\{F\}$  of [Gazdar et al. 85] and means that the value for F is always undefined, i.e. the corresponding category does not take a value for F. The value ' $\rightarrow$ ' is specially treated by the unification and the feature instantiation principles.

introduced for them. The feature specification  $\text{-COH(eren)}$  marks categories that can be extraposed. This metarule fulfils the termination conditions (i) and (iii) and 'SLASH1  $\neg$ '  $>_{\text{OP}}$  'SLASH1 X[-COH]' has to be defined.

**Extraposition Metarule:**

$$\begin{aligned} V^3 \rightarrow V^0, W, X[-\text{COH}] \\ \Rightarrow \\ V^3[\text{SLASH1 X[-COH]}] \rightarrow V^0, W \end{aligned}$$

The metarule for passive is an example in which the termination conditions (ii) and (iii) are necessary, because no category is deleted and an optional prepositional phrase is introduced that replaces the accusative determiner phrase:

**Passive Metarule:**

$$\begin{aligned} V^3[-\text{PAS}] \rightarrow V^0, W, \text{DP}[\text{acc}] \\ \Rightarrow \\ V^3[+\text{PAS}] \rightarrow V^0, W, \text{PP}[\text{von}]. \end{aligned}$$

Here the change of the feature specification of PAS at the mother category can be used for terminating metarule application and we have to define '-PAS'  $>_{\text{OP}}$  '+PAS' and 'DP[acc]'  $>_{\text{OP}}$  'PP[von]'.<sup>4</sup>

The *Auxiliary Metarule* is similar to the *Passive Metarule* in that feature values of some categories are changed and the termination conditions (ii) and (iii) are fulfilled. Here it is the BAR level of the mother and  $V^3$ -daughter that are lowered in analogy to the *Subject Deletion Metarule*. The operator precedence to be defined is 'BAR 3'  $>_{\text{OP}}$  'BAR 2', which already has been defined in connection with that metarule.

**Auxiliary Metarule:**

$$V^3[+\text{AUX}] \rightarrow V^0, V^3 \Rightarrow \text{VP}[+\text{AUX}] \rightarrow V^0, \text{VP}$$

As we have seen, the *Subject Deletion*, the *Slash Termination* and the *Extraposition Metarule* fulfil the criterion of deleting a category on the rhs of the ID rule; the *Passive* and the *Auxiliary Metarule* change feature values at the categories. For all metarules a non-contradictory set of operator precedences can be defined and the application of the whole set of metarules will terminate in every case.

Even the AdvP-metarule in section 3 proposed by [Uszkoreit 87] can be treated when the metarules are applied directly, because we can give a proof for its termination, which is guaranteed by the finite length of the input string in connection with direct application. This is another argument for the direct application of metarules.

## 6 The parsing process

In our parser, which is a part of an experimental machine translation system (see [Weisweber 87] for a

detailed description of the parser without direct application of metarules), the metarules are defined according to the following scheme:

$$C_0 \rightarrow C_c, W, C_d \Rightarrow C'_0 \rightarrow C'_c, W, (C'_d)$$

$C_0$ ,  $C_c$  and  $C_d$  are categories and  $W$  is a variable for a (possibly empty) multiset of categories. The categories  $C'_0$  and  $C'_c$  correspond to  $C_0$  and  $C_c$ , respectively, in terms of [Gazdar et al. 85]. The category  $C_c$  can be viewed as a condition category for the application of the metarule.  $C_d$  is the category which is to be deleted or modified. This is indicated by the brackets around  $C'_d$ . If  $C_d$  is to be deleted,  $C'_d$  is left out on the rhs of the metarule. If  $C_d$  is to be modified,  $C_d$  is replaced by  $C'_d$ . The feature values of the categories are cospecified on the lhs and rhs of a metarule, if not specified otherwise. This causes the values to be carried over to the rhs. If the metarule should only be applied to lexical ID rules as proposed in [Gazdar et al. 85], the category  $C_c$  has to be the lexical head with respect to  $C_0$ .

The proof for the termination of sets of such metarules is simple. At first we look at the case in which  $C'_d$  is deleted, then the termination condition (i) holds.

The second case is that the category  $C_d$  is replaced by the category  $C'_d$  and the number of categories is not reduced. Now the termination condition (ii) has to be applied and at least one feature value of the categories  $\{C_0, C_c\}_m \cup_m W$  has to be changed, which must not be reversed by another metarule.

The termination condition (iii) holds in every case, because the variable  $W$  occurs on both sides of the metarule.

In order to apply the metarules directly during the parsing process, all the categories of an ID rule, which are matched by the multiset  $\{C_c\}_m \cup_m W$ , have to be collected by the parser. This is done for example by the *Completer* of the modified Earley algorithm (see [Earley 70], [Shieber 84], [Kilbury 84] and [Dörre/Momma 85]). Suppose the Completer tries to complete with the inactive edge  $\langle C'_d, i, j, \gamma \rangle_m$ , which is spanning from node  $i$  to node  $j$  of the chart, where  $\gamma$  is a multiset of daughter categories which have already been analysed and the remainder, i.e. the multiset of daughter categories still to be analysed, is empty and  $C'_d$  is the mother category of the ID rule, which is licensing this edge.  $M$  is the set of metarules.

$$\begin{aligned} \text{If } \langle C_0, h, j, \alpha \cdot \beta \rangle \text{ is an inactive edge and} \\ (C_0 \rightarrow C_c, W, C_d \Rightarrow C'_0 \rightarrow C'_c, W) \in M \text{ and} \\ \beta = (C_d)_m \text{ and} \\ C_c \in \alpha \cup_m \beta \end{aligned}$$

then the Completer introduces a new inactive edge  $\langle C'_0, h, j, \alpha \cdot \beta \rangle_m$  and computes its closure. The category  $C_c$  in  $\alpha \cup_m \beta$  is replaced by  $C'_c$ .

$$\begin{aligned} \text{If } \langle C_0, h, i, \alpha \cdot \beta \rangle \text{ is an inactive edge and} \\ (C_0 \rightarrow C_c, W, C_d \Rightarrow C'_0 \rightarrow C'_c, W, C_d) \in M \text{ and} \\ C_c \in \alpha \cup_m \beta \text{ and} \\ C_d \in \beta \text{ and} \\ C'_d \text{ is consistent to the categories in } \beta / (C_d)_m \text{ with} \\ \text{respect to linear precedence} \end{aligned}$$

then the Completer introduces a new edge  $\langle C'_0, h, j, \alpha \cup_m (C'_d)_m \cdot \beta / (C_d)_m \rangle$  and the category  $C_c$  in  $\alpha \cup_m \beta$  is replaced by  $C'_c$ . If the remainder  $\beta / (C_d)_m = \{\}$  then the closure of this edge has to be computed.

The advantage of direct parsing with metarules is an increase of efficiency, because all the inactive edges which are licensed by ID rules introduced by metarules need not to be stored separately and the number of inactive edges generated by the Earley parser is reduced considerably.

<sup>3</sup> Treating passive, one has to say a word about semantics. We do not adopt the semantics of [Gazdar et al. 85] because of its shortcomings (see [Umbach 87]), but developed a semantic level of our own that captures the functor argument structures (FAS, see [Busemann/Haenschild 89] and [Haenschild/Umbach 88]) of sentences and is derived from the syntactic structure via term-rewrite rules. Here an explicit assignment of semantic roles to complements of verbs takes place that is dependent on the subcategorization of the verb and its voice.

<sup>4</sup> In this case we have to define a precedence for all feature values which are changed.

Another interesting approach to direct parsing with metarules, in which the metarules are treated as special kinds of context-free rules, is presented in [Kay 83].

## 7 Conclusion

Metarules are an interesting device to express some important generalizations on phrase structure rules of a natural language grammar. If they are used in preprocessing to compile a huge set of rules out of a small set of basic ones, the parsing process may become very inefficient, because it has to care for the set of basic rules and additionally for the rules which have been derived from them and are very similar to the basic ones. When metarules are applied directly during the parsing process, only the set of basic rules in connection with the metarules have to be considered by the parser. This reduces the set of intermediate solutions (inactive edges) to be stored considerably.

In order to apply metarules directly, it has to be guaranteed that the given set of metarules will terminate if all metarules fulfil the termination criterion in section 4. We gave the termination proof for the metarules of our German grammar in section 5. We think that with the help of this criterion the termination of every relevant set of metarules can be proven, because a metarule is defined to change something in an ID rule, either to delete a category, to modify some feature values, to add a category or to do a combination of all. With the termination criterion it is possible to construct a device which automatically proves the termination of a given set of metarules. This algorithm computes the set of operator precedences from the feature values which are changed on the categories of a metarule.

To enable the parser to process metarules like Uszkoreit's for AdvPs in section 3, which add categories to basic ID rules and for which the termination can be proven, will be subject to future work.

## 8 References

[Barton et al. 87]: G. Edward Barton, Robert C. Berwick and Eric Sven Ristad: "Computational Complexity and Natural Language", MIT Press, Cambridge (Massachusetts), London (England) 1987

[Busemann/Hauenschild 88]: S. Busemann, Ch. Hauenschild: "A Constructive View of GPSG or How to Make it Work", in: Proc. of Coling 88, Budapest 1988, pp. 77-82

[Busemann/Hauenschild 89]: S. Busemann, Ch. Hauenschild: "From FAS Representations to GPSG Structures", in: S. Busemann, Ch. Hauenschild and C. Umbach (eds.): "Views of the Syntax/Semantics Interface", in: Proc. of the Workshop "GPSG and Semantics", KIT-Report 74, Technical University of Berlin 1989, pp. 17-43

[Dershowitz 82]: N. Dershowitz: "Orderings for Term-Rewriting Systems", Theoretical Computer Science 17 (1982), North-Holland, pp. 279-301

[Dershowitz 85]: N. Dershowitz: "Termination", in: G. Goos, J. Hartmanis (eds.): "Rewriting Techniques and Applications", LNCS 202, Dijon, France 1985, pp. 180-224

[Dörre/Momma 85]: J. Dörre, S. Momma: "Modifikationen des Earley-Algorithmus und ihre Verwendung für ID/LP-Grammatiken", Manuscript of the Institute for Computational Language Processing (IMS) at the University of Stuttgart 1985

[Earley 70]: J. Earley: "An Efficient Context-Free Parsing Algorithm", Communications of the ACM 13 (2) 1970, pp. 94-102

[Gazdar et al. 85]: G. Gazdar, E. Klein, G. Pullum and I. Sag: "Generalized Phrase Structure Grammar", Oxford, Blackwell 1985

[Hauenschild/Umbach 88]: Ch. Hauenschild, C. Umbach: "Funktoren-Argument-Struktur, Die satzsemantische Repräsentations- und Transferebene im Projekt KIT-FAST", in: J. Schütz (ed.): "Workshop Semantik und Transfer", EUROTRA-D Working Papers No.6, Saarbrücken 1988, pp. 16-35

[Jacobson 87]: P. Jacobson: Review of [Gazdar et al. 85], in: Linguistics and Philosophy 10, 1987, pp. 389-426

[Kay 83]: M. Kay: "When Meta-rules are not Meta-rules", in: K. Sparck Jones, Y. Wilks (eds.): "Automatic Natural Language Parsing", Ellis Horwood Limited, West Sussex, England 1983, pp. 94-117

[Kilbury 84]: J. Kilbury: "Earley-basierte Algorithmen für direktes Parsen mit ID/LP-Grammatiken", KIT-Report 16, Technical University of Berlin 1984

[Kilbury 86]: J. Kilbury: "Category Cooccurrence Restrictions and the Elimination of Metarules", in: Proc. of Coling 86, Bonn 1986, pp. 50-55

[Shieber 84]: S.M. Shieber: "Direct Parsing of ID/LP Grammars", Linguistics and Philosophy 7 1984, pp. 135-154

[Thompson 82]: H. Thompson: "Handling Metarules in a Parser for GPSG", D.A.I. Research Paper No. 175, University of Edinburgh

[Umbach 87]: C. Umbach: "Zur semantischen Interpretation in der Theorie der GPSG", KIT-IAB 19, Technical University of Berlin 1987.

[Uszkoreit 87]: H. Uszkoreit: "Word Order and Constituent Structure in German", CSLI Lecture Notes Number 8, Stanford University 1987

[Weisweber 87]: W. Weisweber: "Ein Dominanz-Chart Parser für generalisierte Phrasenstrukturgrammatiken", KIT-Report 45, Technical University of Berlin 1987

[Weisweber 89]: W. Weisweber: "Transfer in Machine Translation by Non-Confluent Term-Rewrite Systems", in: Proc. of the 13th German Workshop on Artificial Intelligence (GWA1-89), Eringerfeld, September 1989, pp. 264-269

[Weisweber/Hauenschild 90]: W. Weisweber, Ch. Hauenschild: "A model of Multi-Level Transfer for Machine Translation and Its Partial Realization", KIT-Report 77, Technical University of Berlin 1990 and to appear in: Proc. of the Seminar "Computers & Translation '89", Tiflis 1989

[Weisweber 92]: W. Weisweber: "Term-Rewriting as a Basis for a Uniform Architecture in Machine Translation", in: Proc. of Coling 92, Nantes 1992