Bottom-Up Filtering : a Parsing Strategy for GPSG

Philippe BLACHE (*) and **Jean-Yves MORIN** (**)

*Groupe Représentation et Traitement des Connaissances
CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE
31, Ch. Joseph Aiguier
13402 Marseille Cedex 09 (France)
e-mail : grtc@frmop11.bitnet

** Département de Linguistique et Philologie
UNIVERSITÉ DE MONTRÉAL
CP 6128, Succ. A, H3C 3J7
Montréal (Canada)
e-mail : morinjy@iro.umontreal.ca

Abstract

In this paper, we propose an optimized strategy, called Bottom-Up Filtering, for parsing GPSGs. This strategy is based on a particular, high level, interpretation of GPSGs. It permits a significant reduction of the non-determinism inherent to the rule selection process.

Introduction

Linguistic theories are becoming increasingly important for natural language parsing. In earlier work in this domain, few approaches were based on full-fledged linguistic descriptions. Nowadays, this is becoming the rule rather than the exception[1].

Among all the current linguistic theories, we think that GPSG allows the simplest interface between linguistic and computational theory. But its naive computational interpretation, although fairly straightforward, might result in a computationally expensive implementation. Barton showed that universal ID/LP parsing could be reduced to the vertex-cover problem, and so was NP-complete. In theory, we can only search for heuristics. In actual practice we might still look for efficient implementations. Several authors (Evans, Ristad, Kilbury...) developed an interpretation of the theory that can support an efficient implementation. Some, like [*Shieber86*], are more interested in the algorithmic viewpoint.

In this paper, we shall review some of the most important factors of intractability before giving a presentation of Bottom-Up Filtering. This presentation is twofold: interpretation of GPSG and implementation of this interpretation using Bottom-Up Filtering.

[1] Cf. for instance [*Abramson89*], [*Gazdar89*], [*Uszkoreit88*]. See also [*Jensen88*] for the contrary position.

1. Complexity of GPSG parsing

Several studies like [*Barton84*] or [*Ristad86*] discussed the theoretical complexity of universal GPSG parsing. Here we shall focus on the effective complexity of GPSG parsing and especially on the problem of non-determinism in rule selection.

Rule selection generates several problems more particularly due to local ambiguity: the parser can select a wrong rule and cause backtracking. This non-determinism problem is one of the most important in natural language processing. Several mechanisms such as lookahead or knowledge of leftmost constituents have been proposed to constrain the operation of rule selection and reduce this non determinism.

The ID/LP formalization separates two aspects of phrase structure which are merged in a standard (context-free) phrase structure rule: hierarchical constituency and constituent order. Constituency rules (ID-rules) are represented with unordered right-hand sides. Hence, for an ID-rule like $X \rightarrow A_1, \ldots, A_k$, an unconstrained expansion can generate $k!$ phrase-structure rules. Moreover, metarules increase this overgeneration problem. To summarize this problem, we can say that there are two main sources of non-determinism (and henceforth of actual computational complexity) in GPSG:

(1) ID-rules:
- derivation is a non-deterministic process
- possibility of null transition (rules with empty right-hand sides), permitting large structures with few "supporting" terminals
- unordered right-hand sides

(2) Metarules:
- induction of null transition and ambiguity
- exponential increase of ID-rules
- non-deterministic application to ID-rules

There are several other factors of complexity. Most of the parsing problems come from non-determinism, which can be reduced in two ways: constraints on the underlying linguistic theory and development of new parsing strategies.

2. *Constraining GPSG*

The interpretation of a linguistic theory consists in the adaptation of the abstract model to make it computationally tractable. This adaptation has to be justified both linguistically and computationally.

This notion is quite recent in the domain of natural language parsing systems: most of them use only a small part of the theory, often not in a coherent way, and so introduce many *adhocities*. Moreover, we can give many differents interpretations to the same theory. In the case of GPSG, we can for example interpret it as an independently justified theory or just as a particular formalism for context-free grammars. There is a choice between indirect interpretation (compilation into a context-free grammar, which is then interpreted) and direct interpretation of a GPSG. A compilation of a GSPG consists in several expansion steps which transform it into a context-free equivalent grammar. Compiling a GPSG amounts to considering it as a notational variant of ordinary context-free phrase-structure grammar. As noted by [Evans 87], a direct interpretation is more in keeping with the high level mechanisms of GPSG and might even be actually more efficient than the indirect interpretation approach.

Our interpretation of GPSG does not use a pre-compilation level. It is more particularly oriented towards an adaptation of the ID-rules formalization: problems caused by non-determinism are known to be directly related to grammar representation. In the case of GPSG, these problems arise from the use of unordered rules.

We think that we must respect the high level and generality of GPSG. So, we propose the use of very large ID-rules, called *extensive ID-rules*, able to describe many different constructions for the same phrase. Hence, we partially eliminate subcategorization (whose informational and predictive value has been largely overestimated) and replace it with a Bottom-Up Filtering mechanism.

We propose to apply to GPSG the notion of *automata minimization* exposed in [*Aho72*]. This notion, based on the concept of distinguishability between two states, is used to generate reduced automata (i.e. automata in which no state is unaccessible)[2]. We apply this concept to ID/LP formalization to achieve what we can call *grammar minimization*: no two rules describing the same phrase have more than one common category (the head of the current phrase). In other words, we will use different rules only for very different constructions. Consequently, we never have sets of ID-rules like [*Gazdar85*]:

$$VP \rightarrow H[3], NP, PP[to] \quad (give)$$
$$VP \rightarrow H[4], NP, PP[for] \quad (buy)$$

As for automata, we first have to define equivalence classes for ID-rules. From these classes (also called *families of rules*), we extract a representative element which will be the *extensive ID-rule*. These concepts are formally defined as follows:

- *head of a phrase* : a head h of a phrase P is a constituent with particular properties : its presence is necessary for some constructions of P and moreover, the values of both the N and V features must be the same in the descriptions of the head and of the phrase. Hence, we can define a function *head* from $\mathcal{P}(K \cup \Sigma)$ to K (where K is the set of categories, Σ the set of terminals and $\mathcal{P}(X)$ denotes the set of all subsets of X) as follow : let an ID-rule of the form $A \rightarrow_{di} C_1, \ldots, C_n$, then :

$$head(A) = \{ C_i / (C_i([N]) = A([N])) \text{ and } (C_i([V]) = A([V])) \}$$

let G be a GPSG, R be the set of ID-rules of G, and $r \in R$,
r is of the form $s \rightarrow t, c_i, \ldots, c_j$ (with $0 \leq i \leq j$), where t is the head of s

we define the following operations on R :

[2] This is central for our approach: no two distinct states are indistinguishable.

(1) left-hand side of a rule:
$LHS(r) = \{s\}$
(2) right-hand side of a rule:
$RHS(r) = \{t, ci, \ldots, cj\}$
(3) reduced right-hand side of a rule:
$RHS^{-}(r) = \{ci, \ldots, cj\}$
(4) rule inclusion (noted $\supseteq$)
let $r1, r2 \in R, r1 \supseteq r2$ iff
$LHS(r1) = LHS(r2), head(r1) = head(r2)$
and $RHS^{-}(r1) \supseteq RHS^{-}(r2)$

We define a *rule clustering* function F from R to R as follows : $F(r) = \{r_i \in R / r_i \supseteq r \vee r \supseteq r_i\}$

Hence, an extensive ID-rule is define as follows :

let $r \in R$, r is an ***extensive ID-rule*** iff
$\forall r' \in F(r), r' \supseteq r$

Such a formalization of the grammar considerably reduces the problem of non-determinism during the selection of a rule: if two rules are different, their right-hand sides have at most one element in common. This allows us to establish strong selection constraints. To summarize, using extensive ID-rules allows a very high level of generality for the representation of a GPSG, preserving its succinctness property.

3. *Bottom-Up Filtering*

The Bottom-Up Filtering strategy is based on the detection of the first constituent of a phrase. [*Pereira87*], in a presentation of bottom-up parsing, describes the *left-corner* parsing method. This strategy was first introduced in [*Rosenkrantz70*]. It consists in finding the leftmost constituent α of a phrase P, so as to select a phrase structure rule $P \rightarrow \alpha\ \psi$ and then proving that α is actually the left-corner of such a phrase by application of the rest (ψ) of the selected rule. There are two stages in the process: a bottom-up one (detecting the left corner) and a top-down one (parsing the rest of the phrase). Using both strategies is interesting, particularly for the selection of a phrase structure rule: knowledge of the leftmost constituent constrains this stage and so reduces non-determinism. Hence, this strategy, like ours, is based upon the detection of the leftmost constituent of a phrase. But the similarity stops here: the use of unordered rules, inherent to the ID/LP formalism, would force modification and introduction of new mechanisms. Moreover, this strategy allows only a small reduction of non-determinism, especially because the top-down stage is used in a classical way.

Based on our interpretation of GPSG and the formalization of extensive ID-rules we propose a strategy that allows the initialization of the phrase level upon determination of the leftmost constituents. After this bottom-up stage, the parse is completed by a top-down process consisting in the selection of the adequate extensive ID-rules and the *generation* of phrase-structure rules. We insist on the fact that we don't use expansion or a selection function for this last stage, but a genuine *generation* process: the rules are actually deduced by formal operations from the grammar. This stage is largely constrained by both our formalization and the bottom-up filtering that initializes the phrases. We obtain a strategy in which non-determinism is drastically reduced.

Bottom-Up Filtering parsing is achieved in three stages:

(i) creation of prediction tables
(ii) phrase level initialization
(iii) generation of phrase-structure rules

3.1. *Prediction tables*

Using the extensive ID-rule formalization, we deduce informations that will allow us to determine the leftmost constituent. We use two main concepts: *first legal daughter* and *immediate precedence.*

Definition: *the first legal daughter of a constituent is a category of any level that can occur in the first position of the right-hand side of a phrase-structure rule describing this constituent.*

So, according to LP constraints, a given constituent may have several first legal daughters which we collect into a set.

We note $<$ the linear precedence relation.
Let P be a phrase, $\forall \alpha$ such that $P \rightarrow \alpha$ then *First*, the set of first legal daughters, is defined as follows:

$First(P) = \{c \in \alpha \text{ such that } \forall x \in \alpha - \{c\}, \text{ then } c<x\}$

The second concept, the *immediate precedence* relation, allows us to determine all the constituents that can precede, according with LP constraints, a first legal

daughter in a right-hand side of ID-rule. These constituents can themselves be first legal daughters of the considered phrase, or not. The reason is, particularly when using the extensive ID-rules formalism, that several ID-rules describe several different constructions of a given phrase type. So, there may be constituents that cannot initialize a phrase but, in some constructions, that can precede a constituent which is actually a first daughter in another construction. This relation defines sets of immediate precedence as follows:

Let P be a phrase, $\forall\ \alpha$ such that $P \rightarrow \alpha$, let x be a non-terminal, let $c \in First(P)$, then $IP_P(c)$, the set of immediate precedence of c for P, is defined as follows:

$IP_P(c) = \{x$ such that $(x < c)$ or $(x \in \alpha$ and neither $x < c$ nor $c < x$ exist$)\}$

Prediction tables are made of the sets of first legal daughters for all phrases and those of immediate precedence for each first legal daughter. These sets are specified during the implementation of the grammar. Note that this is not a compilation of the grammar, because we only have to determine the leftmost constituents for the rules, whereas compilation would generate all the possible permutations for entire rules. The sets are thus kept reasonably small.

3.2. *Phrase level initialization*

With the aid of the prediction tables, we can now describe the mechanisms used in the initialization of the phrase level. This consists in determining all the first daughters in the input sentence, and so all the phrases belonging to the syntactic structure. This stage consists in two phases: categorization and actual initialization.

The categorization is a trivial function, used in all bottom-up strategies, which we enhance with a special device for easier resolution of lexical ambiguities: the resulting data are stored as possible backtracking points for our parser.

The initialization stage is based upon a simple principle: an element of the sequence of categories is a first daughter of a phrase if it belongs to the set of first legal daughters of this phrase and if the previous category does not belong to its immediate precedence set. We define the *initialize* relation as follows :

Let G be a GPSG, $L(G)$ the language generated by G, let I be a string such that $I \in L(G)$, let C the list of categories of I, $\forall\ c \in C, \exists\ c' \in N$ such that c' precedes c in C ;

c initialize S iff $c \in First(S)$ and $c' \notin IP_S(c)$

This stage yields a new list made of the lexical categories and the initialized phrases.

We can give a very simple example of phrase level initialization.

Let G a very small ID/LP grammar :

Extensive ID-rules :

S $\rightarrow_{id}$ NP, VP	VP $\rightarrow_{id}$ V, NP, PP
NP $\rightarrow_{id}$ Det, N, AP, PP	AP $\rightarrow_{id}$ Adj
NP $\rightarrow_{id}$ N	PP $\rightarrow_{id}$ Prep, NP

LP-rules (given here in a binary formalization) :

V< NP	V < PP	NP < VP	Det < N
Det < AP	Det < PP	N < SP	Prep < NP

Sets of First Legal Daughter :

First(S) = {NP}	First(NP) = {Det, N}
First(VP) = {V}	First(PP) = {Prep}
First(AP) = {Adj}	

Sets of Immediate Precedence :

IP_S(NP) = ∅	IP_{VP}(V) = ∅
IP_{NP}(Det) = ∅	IP_{NP}(N) = { Det , AP }
IP_{AP}(Adj) = ∅	IP_{PP}(Prep) = ∅

PHRASE LEVEL INITIALIZATION

Let the sentence :

Peter walks down the street.

(1) Categorization :

N . V . Prep . Det . N

(2) Phrase level initialization :

Current cat	*First(P)*	*Precedent cat*	*IPp(c)*	*Action*
N	NP	-	∅	N / NP
V	VP	N	∅	V / VP
Prep	PP	V	∅	Prep / PP
Det	NP	Prep	∅	Det / NP
N	NP	Det	{Det, AP}	-

We obtain the following list :

S . < NP, N > . < VP, V > . < PP, Prep > . < NP, Det > . N

We must keep in mind that the construction of the prediction tables is a pre-processing. The actual step of initialization just consists in applying to the set of lexical

categories the relation as defined before. Hence, this step of our strategy is computationally trivial.

3.3. Phrase-structure rules generation

In this last stage the phrase construction is completed by selecting a pattern ID-rule and then generating the right phrase structure rule.

The pattern ID-rule selection is largely constrained by our formalization using extensive ID-rules, but also by the knowledge of left-hand side and the leftmost element of right-hand side of the rule. It is almost a deterministic process.

The generation stage can only be roughly sketched here. It consists in a top-down search in the list of initialized phrases for the constituents of the current phrase. For each category we scan this list of initialized phrases, adding to the phrase structure rule under generation all the categories belonging to the pattern ID-rule. If a category does not belong to the pattern rule, it can be an indirect constituent (i.e. a category belonging to a constituent itself belonging itself to the phrase which is being parsed). So, we have a process which allows us to generate the phrase structure rules required for the parse.

Conclusion

The Bottom-Up Filtering strategy formalizes theoretical constraints which allow us to reduce the non-determinism problem due to local ambiguities[3]. We have implemented an algorithm based on the Bottom-Up Filtering strategy in Prolog II on a Macintosh SE/30 and obtained interesting results: for a non trivial GPSG of French, most of the analyses for "usual" sentences take less than 1 second. More complicated constructions like passive, coordination or discontinuous constituents take between 1 and 2.5 seconds.

References

[*Abramson89*] Abramson, H. & V. Dahl (1989) *Logic Grammars*, Springer-Verlag.

[*Aho72*] Aho A. & J. Ullman (1972) *The Theory of Parsing, Translation and Compiling, Volume 1: Parsing*, Prentice-Hall.

[*Blache90*] Blache P. (1990) "L'analyse par Filtrage Ascendant : une stratégie efficace pour les Grammaires Syntagmatiques Généralisées", 10th International Workshop Expert Systems and their Applications , Avignon

[*Blaser88*] Blaser, A. (1988 ed.) *Natural Language at the Computer*, Springer-Verlag.

[*Barton84*] Barton G. (1984) "On the complexity of ID/LP parsing", AI Memo # 812, MIT.

[*Barton87*] Barton G., R. Berwick, E. Ristad (1987) *Computational Complexity and Natural Language*, MIT Press.

[*Berwick82*] Berwick R., A. Weinberg (1982) "Parsing Efficiency, Computational Complexity and the Evaluation of Grammatical Theories", *Linguistic Inquiry*, 13, 2.

[*Berwick85*] Berwick R., A. Weinberg (1985) "Deterministic Parsing and Linguistic Explanation", *Language and Cognitive Processes*, 1, 2.

[*Earley70*] Earley J. (1970) "An Efficient Context-Free Parsing Algorithm", *Commnications of the ACM*, 13, 94-102.

[*Evans85*] Evans R. (1985) "ProGram - a development tool for GPSG grammars", *Linguistics*, 23.

[*Evans87*] Evans R. (1987) *Theoretical and Computational Interpretations of GPSG*, Ph. D. Dissertation, University of Sussex.

[*Gazdar85a*] Gazdar G., G. Pullum (1985) "Computationally Relevant Properties of Natural Languages and their Grammars", *New Generation Computing*, 3.

[*Gazdar85b*] Gazdar, G., et al. (1985) *Generalized Phrase Structure Grammar*, Blackwell.

[*Gazdar89*] Gazdar, G. & C. Mellish (1989) *Natural Language Processing in Prolog*, Addison-Wesley.

[*Jensen88*] Jensen, K. (1988) "Issues in Parsing", *in* [Blaser 88]: 65-83.

[*Kilbury88*] Kilbury J. (1988) "Parsing with Category Cooccurrence Restructions", *COLING 88*: 324-327.

[*Morin86*] Morin, J.Y (1986) *Théorie syntaxique et théorie du parsage*, Dép. linguistique & philologie, Univ de Montréal et GIA, Faculté des Sciences de Luminy.

[*Morin89*] Morin, J.Y (1989) *Particules et parsage universel, in* H. Weydt (1989 ed.) *Sprechen mit Partikeln*, Walter de Gruyter (Berlin).

[*Pereira87*] Pereira F., S. Shieber (1987) *Prolog and Natural Language Analysis*, CSLI Lecture Notes # 10.

[*Perrault83*] Perrault C. (1983) "On the Mathematical Properties of Linguistic Theories", *ACL-21*.

[*Phillips86*] Phillips J., H. Thompson (1986) *A Parser for GPSG*, D.A.I Research Paper n° 289, University of Edinburgh.

[*Ristad86*] Ristad E. (1986) "Computational Complexity of Current GPSG Theory", proceedings of ACL.

[*Ristad87*] Ristad E. (1987) "Revised GPSG", *ACL-25*.

[*Rosenkrantz70*] Rosenkrantz D., P. Lewis (1970) "Deterministic Left-Corner Parser", *IEEE Conference record of the 11th Annual Symposium on Switching and Automata Theory*.

[*Shieber84*] Shieber S. (1984) "Direct parsing of ID/LP grammars", *Linguistics and Philosophy*, 7, 135-154.

[*Shieber 86*] Shieber S. (1986) "A Simple Reconstruction of GPSG", *COLING 86*: 211-5.

[*Uszkoreit 88*] Uszkoreit, H. (1988) "From Feature Bundles to Abstract Data Types: New Directions in the Representation and Processing of Linguistic Knowledge", *in* [Blaser 88]: 31-64.

[3] Our phrase level initialization principle is related to the Universal Projection Principle proposed for independent reasons in [*Morin89*].