A PROLOG Implementation of Government-Binding Theory

Robert J. Kuhns

Artificial Intelligence Center
Arthur D. Little, Inc.
Cambridge, MA 02140 USA

Abstract

A parser which is founded on Chomsky's Government-Binding Theory and implemented in PROLOG is described. By focussing on systems of constraints as proposed by this theory, the system is capable of parsing without an elaborate rule set and subcategorization features on lexical items. In addition to the parse, theta, binding, and control relations are determined simultaneously.

1. Introduction

A number of recent research efforts have explicitly grounded parser design on linguistic theory (e.g., Bayer et al. (1985), Berwick and Weinberg (1984), Marcus (1980), Reyle and Frey (1983), and Wehrli (1983)). Although many of these parsers are based on generative grammar, and transformational grammar in particular, with few exceptions (Wehrli (1983)) the modular approach as suggested by this theory has been lagging (Barton (1984)). Moreover, Chomsky (1986) has recently suggested that rule-based parsers are implausible and that parsers could be based on lexical properties and structure determining principles.

This paper describes a principle-based parser which is modular in design and which processes sentences simultaneously with respect to modules of Government-Binding (GB) Theory (Chomsky (1981, 1982, 1986)). This parser requires few grammar rules and no explicit subcategorization features for VPs. We also attempt to show that logic programming (specifically, PROLOG (Clark and Tarnlund (1982), Clocksin and Mellish (1984), Hogger (1984), and Kowalski (1979))) makes perspicuous the principles and constraints which underlie this parser.

2. Overview of Government-Binding Theory

GB-Theory (Chomsky (1981)) has shifted the emphasis of grammar from a system of rules to a system of modules which include:

X-bar
Theta
Case
Bounding
Trace
Control
Binding
Government

For the purposes (and space limitations) of this paper we only briefly describe the theories of X-bar, Theta, Control, and Binding. We also will present three principles, viz., Theta-Criterion, Projection Principle, and Binding Conditions.

2.1 X-Bar Theory

X-bar theory is one part of GB-theory which captures cross-categorial relations and specifies the constraints on underlying structures. The two general schemata of X-bar theory are:

(1)a. $\bar{\bar{X}} \longrightarrow$ Specifier $\bar{X}$

b. $\bar{X} \longrightarrow$ X Complement

The types of categories that may precede or follow a head are similar and Specifier and Complement represent this commonality of the pre-head and post-head categories, respectively.

Although the parse operates in accordance with X-bar theory, it does not require specific instructions for each X (X = N, V, A, P).

2.2 Theta-Theory

Theta-theory is the module which determines a sentence's argument structure and theta or thematic-role (e.g., agency, theme, locative) assignments. It is through theta-relations and general principles that arguments and their possible positions can be predicted and explained.

Theta-roles are assumed to be assigned compositionally, in that a head (i.e., X of an XP = $\bar{\bar{X}}$) assigns a theta-role to its complement and this pair (head and complement) in turn determines the theta-role (if one exists) of its specifiers. For example, in sentences:

(2)a. John broke the bottle.
b. John broke his (own) leg.

BREAK assigns the role of theme to bottle and leg in a. and b., respectively. However, the VP broke the bottle assigns the role of agent to John in a., while broke his leg assigns some other role (perhaps, experiencer) to John in b.

2.3 Empty Categories

One difficulty parsing strategies must solve is the detection of the presence of gaps or empty categories and their antecedents. There are three different sets of properties that may be associated with empty categories (Chomsky (1982)), and these sets determine whether an empty category is a trace, PRO, or a variable. While all of these empty categories are phonologically null, their location and interpretation must be determined for a parse to be complete. In short, a trace remains at an extraction site of Move α, PRO is a pronominal which may be present in ungoverned positions, and variables are Case-marked traces.

2.4 Control Theory

Control theory determines the controller of PRO. In other words, the reference of PRO is derivable by Control theory which assigns an interpretation to PRO as subjects of embedded infinitives:

(3)a. $John_i$ wants [PRO_j to leave].

b. John persuaded $Bill_i$ [PRO_j to leave].

In both (3) a. and b., i=j, but in (3) a. John is the subject, and in b., Bill is the object. In other words, want and persuade are subject and object control verbs, respectively, and are lexically marked as such.

2.5 Binding Theory

Binding theory constrains the assignment of indices (which are interpreted as intended coreference). The binding conditions are:

(4)a. An anaphor is bound in its governing category.
b. A pronominal is free in its governing category.
c. An R-expression is free.

An R-expression is a referential term such as a proper noun or a variable. A governing category is the minimal S or NP which contains an anaphor or pronominal and a governor of that anaphor or pronominal. And X is a governor of Y iff X = A, N, V, or P and Y is contained in the smallest maximal projection of X (i.e., the smallest XP) and X c-commands Y. C-command is defined in the usual way, that is, X c-commands Y iff the first branching node dominating X also dominates Y, and X does not dominate Y.

2.6 Chains, Theta-Criterion, and Projection Principle

Intuitively, a chain encodes the history of movement of a constituent. We distinguish between two landing sites of movement, namely, an argument position (A-position) and a non-argument position ($\bar{A}$-position). NP-movement moves or relates a gap with another A-position within an S while wh-movement relates a position in an S to a position in COMP, which is outside of S and is an $\bar{A}$-position. We will limit our discussion to A-positions.

Definition. A chain ($\alpha_1, \ldots, \alpha_n$) is a sequence consisting of a head (α_1) and locally bound traces $\alpha_2, \ldots, \alpha_n$.

Definition. A locally binds B iff either A is the nearest head binding B or A is a locally bound trace which is the nearest binder of B.

It should be noted that all arguments must be in one and only one chain. It is argued in GB-theory that both Case and theta-roles are assigned to chains rather than individual NPs. Theta-roles are assigned according to a strict condition called the Theta-criterion.

(5) Each chain receives one and only one theta-role.

This says basically that theta-role assignments are complete and well-defined.

The question of where in a grammar the Theta-criterion holds is answered by the Projection Principle.

(6) The Theta-criterion is satisfied at all levels of syntactic representation, namely, D-structure, S-structure, and LF (logical form).

We exploit the notions of chains, and principles (5) and (6) in our system. Since a head theta-marks its complement as specified in the lexicon, the force of (5) and (6) is that D-structure, S-structure, and LF are projections from the lexicon.

3. Modules of the Parser

The parser processes a sentence and outputs a triple whose parts are simultaneously determined and consists of a constituent analysis, intended coreference relations (binding and control), and argument structures (theta-relations). Since a distinguishing fèature of this parser is the processing of the latter two representations, we will discuss only the derivations of them.

It should be noted that, although the structural analysis of the parse will not be presented in this paper, the parser is a deterministic one with a limited look-ahead facility (Marcus (1980)). In essence, it is deterministic in that all structures created are permanent and cannot be modified or deleted. In other words, the structures created during the parse are equivalent to the structures of the output of the parse.

The next two subsections will sketch the lexical component and the scope of the grammar. Binding, control, and theta conditions will be presented in Sections 4. and 5.

3.1 Lexicon

The lexicon is a critical component; it contains all the processable words and their associated syntactic and semantic features. The syntactic characterization includes X-bar features (±N, ±V), tense, number, etc.

Traditionally, the features also contain subcategorizations or templates which specify the types of complements (if any) a lexical entry could take. For instance, a subcategorization would indicate whether or not a verb is transitive. However, these templates are redundant in that we can replace them with the theta-roles which an entry (e.g., a verb) assigns to or theta-marks its complement. From this, the parser derives the subcategorization. For instance, the verb told selects a goal and a proposition. A goal is structurally realized as an NP and a proposition must be either an S or an NP. The choice between the structure of S or NP is determinable given a particular S as input.

3.2 Grammar Rules

Incorporating GB theory into the parser helps to eliminate many grammar rules because of their redundancy. As seen above, syntactic structure is derivable from means other than explicit rules. The parser does require a set of grammar rules and we hope to reduce this set in later versions. It should be noted that since priority during implementation was given to Binding theory, Theta-theory, and chains, some rules were used for ease of development. As mentioned above, we plan to eliminate rules which are unnecessary because the structures they specify can be derived from other general principles. However, some rules which describe language-specific properties or marked structures may be necessary and, thus, will have to be stated explicitly.

Some of the rules the parser presently needs are those that deal with NP constructions. The rule S→NP INFL VP is used as well as some specific rules for determining imperatives and interrogatives (e.g., subject-auxiliary inversion).

We are using rule to mean a phrase structure rule (e.g., a familiar rewriting rule or an X-bar schema) within a grammar. Rule can also denote an implementation of the above concept, i.e., a production rule or a PROLOG clause. The choice of interpretation should be clear from context.

As contrasted with rules, principles are general constraints on syntactic representations (and not on rule application as could be argued). The significance of principles is to constrain the class of possible syntactic representations. The Projection Principle (6), for instance, severely restricts the argument structure of D-structure, S-structure, and LF. This bound on syntactic representation enables a parser to predict syntactic structure without explicit rules.

4. Implementation Considerations

The next several sections will focus on the conceptual overview of the processors involved in our system in addition to fragments of a PROLOG implementation of certain aspects of the system.

4.1 The Interpreter

Similar to Marcus (1980), the basic data structures of this parser are two lists whose elements are represented as terms of predicates. One list (INPUT-BUFFER) is for input and the other (PROCESSED-NODES) is for the (partially) processed nodes or subtrees. These two lists are viewed as changing states rather than pushing and popping stacks. This approach seems reasonable since the parser is not relying on production-like grammar rules.

Although there are lower-level operations or predicates, e.g., LABEL, which labels nodes with features, the basic predicates which are central are CREATE-NODE and INSERT. CREATE-NODE will construct a new node of a pre-specified type and attach it to a child of a particular node. INSERT will insert a specific lexical item, a trace, or a PRO as appropriate. Since the output that represents the structure is the familiar labelled bracketing, these predicates do call list manipulation predicates.

It should be noted that many of the tree-walking algorithms that are needed to examine terms of PROCESSED-NODES can be succinctly specified while the underlying unification/ resolution components of PROLOG produce the necessary tree walk.

4.2 Grammar Interface

As noted above, the parser is constrained by X-bar theory. So, if a specifier of a category is the first term of INPUT-BUFFER, then by schema (1)a. the parser creates (using CREATE-NODE) first an XP, and then the specifier. The X-bar features specified in the lexicon determine the type of XP. Similarly, (1)b. will determine when the parser is to create an X node and a complement.

Since all XPs must contain a head, a predicate CREATE-HEAD is a separate module.

4.3 Indexing

Binding theory (4)a.-c. is represented as an indexing scheme on the bracketed structure being generated by the parser. In order to illustrate the main ideas, the heads of underlying lower-level predicates will only be described without their bodies. The predicates PARENT-OF (?child, ?parent, ?structure) and DOMINATE (?node1, ?node2, ?structure) are fairly obvious in that in the former ?parent is the node immediately dominating ?child in some tree (?structure). DOMINATE states that ?node1 is dominated by ?node2 in ?structure.

It should be emphasized that Binding Theory can apply only after structure has been built. So ?structure in both predicates refers to the tree in PROCESSED-NODES.

BRANCHING-NODE, FIRST-BRANCHING-NODE, and C-COMMAND are defined in the obvious way. With the assumption that only S and NP are cyclic nodes, the PROLOG representations of these facts are CYCLIC-NODE (S) and CYCLIC-NODE (NP). These predicates are used to define Governing-Category.

Binding theory can now be clearly expressed as:

```
(7)a. BINDING-THEORY (?argument,
        ?structure):—
        ANAPHOR (?argument)
        GOVERNING-CATEGORY (?gov-cat,
          ?argument, ?structure)
        BOUND (?gov-cat, ?argument,
          ?structure)

   b. BINDING-THEORY (?argument,
        ?structure):—
        PRONOMINAL (?argument)
        GOVERNING-CATEGORY (?gov-cat,
          ?argument, ?structure)
        FREE (?gov-cat, ?argument,
          ?structure)

   c. BINDING-THEORY (?argument,
        ?structure):—
        R-EXPRESSION (?argument)
        ABSOLUTE-FREE (?sentence, ?argument,
          ?structure).
```

BOUND, FREE, and ABSOLUTE-FREE are the predicates which have access to PROCESSED-NODES and they specify as to whether or not two indices are to be unified. BOUND will ensure two indices are identical and FREE and ABSOLUTE-FREE will do otherwise. The PROLOG statements (7)a.-c. are a natural expression of (4)a.-c.

4.4 Chains

The process by which chains are constructed and theta-roles assigned will be illustrated in the next section. The notion of chain and local binding can easily be formalized as:

```
(8)a. CHAIN (   ).
   b. CHAIN (?N):— Head (?N).
   c. CHAIN (?N1, ?N2,...,?NK):—
        LOCAL-BIND (?N1, ?N2)
        CHAIN (?N2...?NK).

(9)a. LOCAL-BIND (?N1, ?N2):—
        HEAD (?N1)
        NEAREST-BINDER (?N1, ?N2).

   b. LOCAL-BIND (?N1, ?N2):—
        TRACE (?N1)
        NEAREST-BINDER (?N1, ?N2).
```

For expository reasons, the sequence processing predicates have been suppressed and notation abused. However, NEAREST-BINDER where the first term binds the second will involve C-COMMAND and locality constraints. A chain consists of either a head ((8)b.) or a sequence consisting of one head (?N1) and one or more traces (?N2,...,?NK). The local binding condition in the definition can be captured naturally by the recursive call in (8)c. The clause in (8)a. is the exit of the recursion.

5. Two Examples of the Parsing Strategy

This section will provide two overlapping examples to illustrate the strategy the parser uses to interface with the various modules of GB-theory in order to arrive at a final parse complete with indexing and theta-relations.

Suppose the input to the parser is the sentence:

(10) The instructor told the students to leave early.

The parser first constructs the NP the instructor and then encounters the verb told. It determines (from the lexicon) the theta-roles assigned by told to its complements. In this case, the theta-roles are goal and proposition. As discussed above, a component of the parser infers the constituent structure of the categories marked by a verb. Thus, the system determines that there ought to be an NP adjacent to told in (10) (otherwise, it inserts a trace in that position) followed by an NP or S. With its limited look-ahead capability, the parser sees the two items to and the verb leave. It then knows the realization (viz., S) of the second object and is able to complete the VP and, consequently, the parse.

In order to see the interactions of theta-relations, Binding conditions, and Control theory consider the sentence.

(11) The students were told to leave early.

Suppressing unnecessary details, we construct the various representations of the parse as (11) is processed.

As the students is labelled, it is pushed onto a chain CHAIN-1, and assigned an index. With the verb told being passivized, i.e., in the environment of were, the parser will detect a gap. As in (10) the parser determines (from its theta-markings) that two objects are required for told. With no explicit NP object of told present, it inserts a trace in the parsed tree and pushes the trace onto CHAIN-1 and assigns CHAIN-1 the theta-role of theme (this role is the role which told theta-marks its first object). The parser invokes principle (4)a. (i.e., (7)a.) of Binding Theory and co-indexes the students and trace. CHAIN-1 is now complete because CHAIN-1 is assigned one (and only one) theta-role.

Note that while this parser has a limited look-ahead, it is able to look at all partial structures it has created (although it cannot alter any of them). In this way, this parser can determine local bindings as it processes. Thus, in this case, the parser knows that the NP the students locally binds the trace after told and CHAIN-1 is well-formed.

Again, as in (10) the parser determines the existence of an S and creates PRO as the subject of the embedded infinitive. It pushes PRO onto a new CHAIN-2 and later assigns it the role of agent. The parser also equates the indices of CHAIN-1 and CHAIN-2 because told is an object control verb and the parser already knows the index of the trace. In this way, Control theory is maintained and the correct referential relations hold. The parse is completed in the usual manner.

With the construction of chains and theta-role assignment, we are able to arrive at a (formal) semantic relation while parsing, but unlike Marcus (1980), it is based on a principled, linguistically-based representation of arguments. Also, the binding relations are computed when sufficient information is present to comply with Binding or Control theory.

6. Syntactic Scope and Implementation Issues

The parser has a wide coverage of syntactic structure. It is capable of determining gaps in (multiple) wh-movements. For instance, in

(12) Who [did Bill think [t [the doctor treated t]]]

there are two gaps, one in COMP and the other in the object position of treated. The latter empty category is determinable as in Section 5.

However, the trace in COMP is inferred (using Bounding Theory or subjacency conditions, which restrict distance between landing and extraction sites of movement) because who is in a COMP position and must bind a variable. However, this binding relation cannot be "too far" and so local binders can be constructed when an S is encountered before a Case-marked trace (i.e., a variable) is. In (12) we see that the last trace is the variable which is ultimately bound by who, but subjacency requires a local binder and it must be in an $\bar{A}$ position. Thus, the trace in COMP is inserted, although the variable is not yet visible to the parser.

A fuller account of theta representations is also being developed in that although chains are constructed, the theta relations among chains must be obtained. In (2)a. there are two chains (John) and (the bottle) and in (2)b. the chains are (John) and (his leg). However, it is the verb together with the chains (the bottle) and (his leg) which determine the theta-role of (John). This requires a more substantive account of theta-theory than is currently available in the literature.

Some time is being spent in extending the parser to process parasitic gaps, to determine the cases where pronouns behave as variables, and to determine quantificational relations (Cushing (1982, 1983)).

7. Conclusions

We believe that a modular parser grounded on GB theory, a theory of linguistic subsystems, is feasible and significant in that it sheds light on how a theory of competence may be embedded in one aspect of language use, namely, parsing. Moreover, the strategy we are pursuing is to exploit the interfaces of GB subtheories which seem to allow simultaneous processing of syntactic structure, theta-relations, and binding conditions. This may help to explain the rapidity of human sentence understanding.

8. Acknowledgements

I would like to thank Steven Cushing, Daniel Sullivan, and Mary Zickefoose for reading and commenting on an earlier draft of this paper.

9. References

Barton, Jr., G.E., (1984), "Toward a Principled-Based Parser," A.I. Memo No. 788, MIT, Cambridge, MA.

Bayer, S., L. Joseph, and C. Kalish, (1985), "Grammatical Relations as the Basis for Natural Language Parsing and Text Understanding," Proc. of IJCAI-85, Los Angeles, CA, pp. 788-790.

Berwick, R.C., and A.S. Weinberg, (1984), The Grammatical Basis of Linguistic Performance, The MIT Press, Cambridge, MA.

Chomsky, N., (1981), Lectures on Government and Binding, Foris Publications, Dordrecht-Holland.

Chomsky, N., (1982), Some Concepts and Consequences of the Theory of Government and Binding, The MIT Press, Cambridge, MA.

Chomsky, N., (1986), Knowledge of Language, Praeger, New York, NY.

Clark, K.L., and S.-A. Tarnlund, (1982), Logic Programming, Academic Press, New York, NY.

Clocksin, W.F., and C.S. Mellish, (1984), Programming in Prolog, Springer-Verlag, Berlin.

Cushing, S., (1982), Quantifier Meanings: A Study in the Dimensions of Semantic Competence, North-Holland, Amsterdam.

Cushing, S., (1983), "Abstract Control Structures and the Semantics of Quantifiers," Proceedings of the First Conference of the European Chapter of the Association for Computational Linguistics, Pisa, Italy.

Hogger, C.J., (1984), Introduction to Logic Programming, Academic Press, New York, NY.

Kowalski, R.A., (1979), Logic for Problem Solving, Elsevier Science Publishing Co., Inc., New York, NY.

Marcus, M., (1980), A Theory of Syntactic Recognition for Natural Language, The MIT Press, Cambridge, MA.

Reyle, V., and W. Frey, (1983), "A PROLOG Implementation of Lexical Functional Grammar," Proc. of IJCAI-83, Karlsruhe, West Germany, pp. 693-695.

Wehrli, E., (1983), "A Modular Parser for French," Proc. of IJCAI-83, Karlsruhe, West Germany, pp. 686-689.