

An Attribute-Grammar Implementation of Government-binding Theory

Nelson Correa

Department of Electrical and Computer Engineering

Syracuse University

111 Link Hall

Syracuse, NY 13244

ABSTRACT

The syntactic analysis of languages with respect to Government-binding (GB) grammar is a problem that has received relatively little attention until recently. This paper describes an attribute grammar specification of the Government-binding theory. The paper focuses on the description of the attribution rules responsible for determining antecedent-trace relations in phrase-structure trees, and on some theoretical implications of those rules for the GB model. The specification relies on a transformation-less variant of Government-binding theory, briefly discussed by Chomsky (1981), in which the rule move-a is replaced by an interpretive rule. Here the interpretive rule is specified by means of attribution rules. The attribute grammar is currently being used to write an English parser which embodies the principles of GB theory. The parsing strategy and attribute evaluation scheme are cursorily described at the end of the paper.

Introduction

In this paper we consider the use of attribute grammars (Knuth, 1968; Waite and Goos, 1984) to provide a computational definition of the Government-binding theory layed out by Chomsky (1981, 1982). This research thus constitutes a move in the direction of seeking specific mechanisms and realizations of universal grammar. The attribute grammar provides a specification at a level intermediate between the abstract principles of GB theory and the particular automata that may be used for parsing or generation of the language described by the theory. Almost by necessity and the nature of the goal set out, there will be several arbitrary decisions and details of realization that are not dictated by any particular linguistic or psychological facts, but perhaps only by matters of style and possible computational efficiency considerations in the final product. It is therefore safe to assume that the particular attribute grammar that will be arrived at admits of a large number of non-isomorphic variants, none of which is to be preferred over the others *a priori*. The specification given here is for

English. Similar specifications of the parametrized grammars of typologically different languages may eventually lead to substantive generalizations about the computational mechanisms employed in natural languages.

The purpose of this research is twofold: First, to provide a precise computational definition of Government-binding theory, as its core ideas are generally understood. We thus begin to provide an answer to criticisms that have recently been leveled against the theory regarding its lack of formal explicitness (Gazdar et al., 1985; Pullum, 1985). Unlike earlier computational models of GB theory, such as that of Berwick and Weinberg (1984), which assumes Marcus' (1980) parsing automaton, the attribute grammar specification is more abstract and neutral regarding the choice of parsing automata. Attribute grammar offers a language specification framework whose formal properties are generally well-understood and explored. A second and more important purpose of the present research is to provide an alternate and mechanistic characterization of the principles of universal grammar. To the extent that the implementation is correct, the principles may be shown to follow from the system of attributes in the grammar and the attribution rules that define their values.

The current version of the attribute grammar is presently being used to implement an English parser written in Prolog. Although the parser is not yet complete, we expect that its breath of coverage of the language will be substantially larger than that of other Government-binding parsers recently reported in the literature (Kashket (1986), Kuhns (1986), Sharp (1985), and Wehrli (1984)). Since the parser is firmly based on Government-binding theory, we expect its ability to handle natural language phenomena to be limited only by the accuracy and correctness of the underlying theory.

In the development below I will assume that the reader is familiar with the basic concepts and terminology of Government-binding theory, as well as with attribute grammars. The reader is referred to Sells (1985) for a good introduction to the

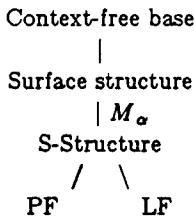
relevant concepts of GB theory, and to Waite and Goos (1984) for a concise presentation on attribute grammars.

The Grammatical Model Assumed

For the attribute grammar specification we assume a transformation-less variant of Government-binding theory, briefly discussed by Chomsky (1981, p.89-92), in which rule move- α is eliminated in favor of a system M_α of interpretive rules which determines antecedent-trace relations. A more explicit proposal of a similar nature is also made by Koster (1978). We assume a context-free base, satisfying the principles of X'-theory, which generates directly structure trees at a *surface structure* level of representation. S-structure may be derived from surface structure by application of M_α . The rest of the theory remains as in standard Government-binding (except for some obvious reformulation of principles that refer to Grammatical Functions at D-Structure).

The grammatical model that obtains is that of (1). The base generates surface structures, with phrases in their surface places along with empty categories where appropriate. Surface structure is identical to S-structure, except for the fact that the association between moved phrases and their traces is not present; chain indices that reveal history of movement in the transformational account are not present. The interpretive system M_α , here defined by attribution rules, then applies to construct the absent chains and thus establish the linking relations between arguments and positions in the argument structures of their predicates, yielding the S-structure level. In this manner the operations formerly carried out by transformations reduce to attribute computations on phrase-structure trees.

(1)



Interpretive Rule

I sketch briefly how the interpretive system M_α is defined. Two attributes *node* and *Chain* are associated with NP, and a method for functionally classifying empty categories in structure trees is

developed (relying on conditions of Government and Case-marking). In addition, two attributes *A-Chain* and \bar{A} -*Chain* are defined for every syntactic category which may be found in the c-command domain of NP. In particular, *A-Chain* and \bar{A} -*Chain* are defined for C, COMP', S, INFL', VP, and V' (assuming Chomsky's (1986) two-level X'-system). The meanings attached to these attributes are as follows. *Node* defines a preorder enumeration of tree nodes; *Chain* is an integer that represents the syntactic chain to which an NP belongs; *A-Chain* (\bar{A} -*Chain*) determines whether an argument (non-argument) chain propagates across a given node of a tree, and gives the number of that chain, if any.

Somewhat arbitrarily, and for the sake of concreteness, we assume that a chain is identified by the *node* number of the phrase that heads the chain.

For the root node, the attribution rules dictate $A\text{-}Chain} = \bar{A}\text{-}Chain} = 0$. The two attributes are then essentially percolated downwards. However, whenever a lexical NP or PRO is found in a θ -position, an argument chain is started, setting the value of *A-Chain* to the *node* number of the NP found, which is used to identify the new chain. Thus NP traces in the c-command domain of the NP are able to identify their antecedent. Similarly, when a Wh-phrase is found in COMP specifier position, the value of $\bar{A}\text{-}Chain$ is set to the chain number of that phrase, and lower Wh-traces may pick up their antecedent in a similar fashion.

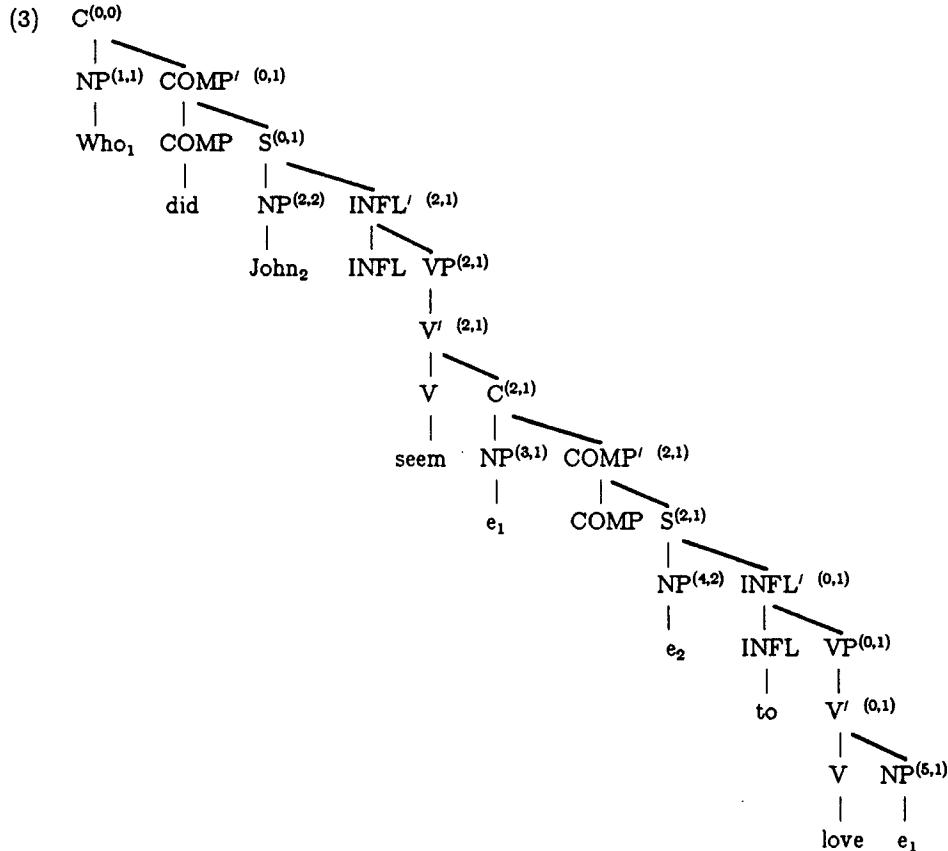
Downwards propagation of the attributes *A-Chain* and \bar{A} -*Chain* explains in a simple way the observed c-command constraint between a trace and its antecedent.

The precise statement of the attribution rules that implement the interpretive rule described is given in Appendix A. In the formulation of the attribution rules, it is assumed that certain other components of Government-binding theory have already been implemented, in particular parts of Government and Case theories, which contribute to the functional determination of empty categories. The implementation of the relevant parts of these subtheories is described elsewhere (Correa, in preparation). We assume that all empty categories are base-generated, as instances of the same EC [$NP \ e$]. Their types are then determined structurally, in manner similar to the proposal made by Koster (1978). The attributes *empty*, *pronominal*, and *anaphoric* used by the interpretive system achieve a full functional partitioning of NP types (van Riemsdijk and Williams (1986), p.278); their

values are defined by attribution rules in Appendix B, relying on the values of the attributes *Governor* and *Cases*. The values of these attributes are in turn determined by the Government and Case theories, respectively, and indicate the relevant governor of the NP and grammatical Case assigned to it.

The claim associated with the interpretive rule, as it is implemented in Appendix A, is that given a *surface structure* in the sense defined above, it will derive the correct antecedent-trace relations after it applies. An illustrative sample of its operation is provided in (3), where the (simplified) structure tree of sentence (2) is shown. The annotations superscripted to the C, COMP', S, INFL', VP, and V' nodes are the *A-Chain* and *A'-Chain* attributes, respectively. Thus, for the root node, the value of both attributes is zero. Similarly, the superscripts on the NP nodes represent the *node* and *Chain* attributes of the NP. The last NP in the tree, complement of 'love', thus bears *node* number 5 and belongs to *Chain* 1.

- (2) *Who_i*, did *John_j*, seem [*e_i*, [*e_j* to love *e_i*]]



Some Theoretical Implications: Bounding Nodes and Subjacency

In Government-binding theory it is assumed that the set of bounding nodes that a language may select is not fixed across human languages, but is open to parametric variation. Rizzi (1978) observed that in Italian the Subjacency condition is systematically violated by double Wh-extraction constructions, as in (4.a), if one assumes for Italian the same set of bounding nodes as for English. The analogous construction (4.b) is also possible in Spanish. A solution, considered by Rizzi to explain the grammaticality of (4), is to assume that in Italian and Spanish, COMP specifier position may be "doubly filled" in the course of a transformational derivation, while requiring that it be not doubly filled (by non-empty phrases) at S-Structure. Thus both moved phrases 'a cui' and 'che storie' can move to the lowest COMP position in the first transformational cycle, while in the second cycle 'a cui' may move to the next higher COMP and 'che storie' stays in the first COMP.

A second solution, which is the one adopted by Rizzi and constitutes the currently accepted explanation of the (apparent) Subjacency violation, is to assume that Italian and Spanish select C and NP as bounding nodes, a set different from that of English. The first phrase 'che storie' may then move to the lowest COMP position in the first transformational cycle, while the second, 'a cui', moves in the next cycle in one step to the next higher position, crossing two S nodes but, crucially, only one C node. Thus Subjacency is satisfied if C, not S, is taken as a bounding node.

- (4) a. Tuo fratello, [a cui], mi domando [che storie], abbiano raccontato e_i , e_i , era molto preoccupato.

Your brother, to whom I wonder what stories they have told, was very worried.

- b. Tu hermano, [a quien], me preguntó [que historias], le habrían contado e_i , e_i , estaba muy preocupado.

The empirical data that arguably distinguishes between the two proposed solutions is (5.a). While the "doubly filled" COMP hypothesis allows indefinitely long Wh-chains with doubly filled COMPs, making it possible for a Wh-chain element and its successor to skip more than one COMP position that already contains some Wh-phrase, the "bounding node" hypothesis states that at most one filled COMP position may be skipped. Thus, the second hypothesis, but not the first, correctly predicts the ungrammaticality of (5.a).

- (5) a. * Juan, [a quien], no me imagino [cuanta gente], e_i , sabe *donde_t*; han mandado e_i , e_t , desaparecio ayer.

Juan, whom I can't imagine how many people know where they have sent, disappeared yesterday.

- b. La Gorgona, [a donde], no me imagino [cuanta gente], e_i , sabe [a quienes], han mandado e_t , e_i , es una bella isla.

La Gorgona, to where I can't imagine how many people know whom they have sent, is a beautiful island.

One might observe, however, that (5.a), even if it satisfies subjacency, violates Pesetsky's (1982) Path Containment Condition (PCC). Thus, on these grounds, (5.a) does not decide between the two hypotheses. The grammaticality of (5.b), on the other hand, which is structurally similar to (5.a) but satisfies the PCC, argues in favor of the "doubly filled" COMP hypothesis. The Wh-phrase 'a donde' moves from its D-Structure position to the surface

position, skipping two intermediate COMP positions. This is possible if we assume the doubly filled COMP hypothesis, and would violate Subjacency under the alternate hypothesis, even if C is taken as the bounding node. We expect a similar pattern (5.b) to be also valid in Italian.

Movement across doubly filled COMP nodes, satisfying Pesetsky's (1982) Path Containment Condition, may be explained computationally if we assume that the type of the \bar{A} -Chain attribute on chain nodes is a last-in/first-out (lifo) stack of integers, into which the integers identifying \bar{A} -chain heads are pushed as they are first encountered, and from which chain identifiers are dropped as the chains are terminated. If we further assume that the type of the attribute is universal, we may explain the typological difference between Italian and English, as it refers to the Subjacency condition, by assuming the presence of an \bar{A} -Chain stack depth bound, which is parametrized by universal grammar, and has the values 1 for English, and 2 (or possibly more) for Italian and Spanish.

To conclude this section, it is worth to review the manner in which the subjacency facts are explained by the present attribute grammar implementation. Notice first that there is no particular set of categories in the theory that have been declared as Bounding categories. There is no special procedure that checks that the Subjacency condition is actually satisfied by, say, traversing paths between adjacent chain elements in a tree and counting bounding nodes. Instead, the facts follow from the attribution rules that determine the values of the attributes A -Chain and \bar{A} -Chain. This can be verified by inspection of the possible cases of movement.

Thus, NP-movement is from object or INFL specifier position to the nearest INFL specifier which c-commands the extraction site. Similarly, Wh-movement is from object, INFL specifier, or COMP specifier position to the nearest c-commanding COMP specifier. If the bound on the depth of the \bar{A} -Chain stack is 1, either S or COMP' (but not both) may be taken as bounding node, and Wh-island phenomena are observable. If the bound is 2 or greater, then C is the closest approximation to a bounding node (although cf. (5.b)), and Wh-island violations which satisfy the PCC are possible. NP is a bounding node as a consequence of the strong condition that no chain spans across an NP node, which in turn is a consequence of the rules (ii.e) in Appendix A.

Parser Implementation

A prototype of the English parser is currently being developed using the Prolog logic programming language. As mentioned in the introduction, the attribute grammar specification is neutral regarding the choice of parsing automaton. Thus, several suitable parser construction techniques (Aho and Ullman, 1972) may be used to derive a parser. The context-free base used by the attribute grammar is an X'-grammar, essentially as in Jackendoff (1977), although some modifications have been made. In particular, following Chomsky (1986) we assume that maximal projections have uniformly bar-level 2 and that S is a projection of INFL, not V, as Jackendoff assumes. The base, due to left-recursion in several productions, is not LR(k), for any k.

We have developed a parser which is essentially LL(1), and incorporates a stack depth bound which is linearly related to the length of the input string. Prolog's backtracking mechanism provides the means for obtaining alternate parses of syntactically ambiguous sentences. The parser performs reasonably well with a good number of constructions and, due to the stack bound, avoids potentially infinite derivations which could arise due to the application of mutually recursive rules. Attributes are implemented by logical variables which are associated with tree nodes (cf. Arbab, 1986). Most attributes can be evaluated in a preorder traversal of the parse tree, and thus attribute evaluation may be combined with LL(1) parser actions. Notable exceptions to this evaluation order are the attributes *Governor*, *Cases*, and θ_s associated with the NP in INFL specifier position. The value of these attributes cannot be determined until the main verb of the relevant clause is found.

Conclusions

We have presented a computational specification of a fragment of Government-binding theory with potentially far-reaching theoretical and practical implications. From a theoretical point of view, the present attribute grammar specification offers a fairly concrete framework which may be used to study the development and stable state of human linguistic competence. From a more practical point of view, the attribute grammar serves as a starting point for the development of high quality parsers for natural languages. To the extent that the specification is explanatorily adequate, the language described by the grammar (recognized by the parser) may be changed by altering the values of the universal parameters in the grammar and

changing the underlying lexicon.

Acknowledgements

I would like to thank my dissertation advisor, Jaklin Kornfilt, for helpful and timely advise at all stages of this research. Also, I wish to thank an anonymous ACL reviewer who pointed out the similarity of the grammatical model I assume to that proposed by Koster (1978), Mary Laughren and Beth Levin for their discussion and commentary on related aspects of this research, Ed Barton, who kindly made available some of the early literature on GB parsing, Mike Kashket for some critical comments, and Ed Stabler for his continued support of this project. Support for this research has been provided in part by the CASE Center at Syracuse University.

References

- Aho, A.V., and J.D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*. Prentice-Hall, Englewood Cliffs, NJ
- Arbab, Bijan. 1986. "Compiling Circular Attribute Grammars into Prolog." *IBM Journal of Research and Development*, Vol. 30, No. 3, May 1986
- Berwick, Robert and Amy Weinberg. 1984. *The Grammatical Basis of Linguistic Performance*. The MIT Press, Cambridge, MA
- Chomsky, Noam. 1981. *Lectures on Government and Binding*. Foris Publications, Dordrecht
- Chomsky, Noam. 1982. *Some Concepts and Consequences of the Theory of Government and Binding*. The MIT Press, Cambridge, MA
- Chomsky, Noam. 1986. *Barriers*. The MIT Press, Cambridge, MA
- Correa, Nelson. In preparation. *Syntactic Analysis of English with respect to Government-binding Grammar*. Ph.D. Dissertation, Syracuse University
- Gazdar, Gerald, Ewin Klein, Geoffrey Pullum, and Ivan Sag. 1985. *Generalized Phrase Structure Grammar*. Harvard University Press, Cambridge, MA
- Jackendoff, Ray. 1977. *X Syntax: A Study of Phrase Structure*. The MIT Press, Cambridge, MA
- Kashket, Michael. 1986. "Parsing a Free-word Order Language: Walpiri." *Proceedings of the 24th Annual Meeting of the Association for*

Computational Linguistics, p.60-66.

- Knuth, Donald E. 1968. "Semantics of Context-free Languages." In *Mathematical Systems Theory*, Vol. 2, No. 2, 1968
- Koster, Jan. 1978. "Conditions, Empty Nodes, and Markedness." *Linguistic Inquiry*, Vol. 9, No. 4.
- Kuhns, Robert. 1986. "A PROLOG Implementation of Government-binding Theory." *Proceedings of the Annual Conference of the European Chapter of the Association for Computational Linguistics*, p.546-550.
- Marcus, Mitchell. 1980. *A Theory of Syntactic Recognition for Natural Language*. The MIT Press. Cambridge, MA
- Pesetsky, D. 1982. *Paths and Categories*. Ph.D. Dissertation, MIT
- Pullum, Geoffrey. 1985. "Assuming Some Version of the X-bar Theory." Syntax Research Center, University of California, Santa Cruz
- Rizzi, Luigi. 1978. "Violations of the Wh-Island Constraint in Italian and the Subadjacency Condition." *Montreal Working Papers in Linguistics* 11
- Sells, Peter. 1985. *Lectures on Contemporary Syntactic Theories*. Chicago University Press. Chicago, Illinois
- Sharp, Randall M. 1985. *A Model of Grammar Based on Principles of Government and Binding*. M.Sc Thesis, Department of Computer Science, University of British Columbia. October, 1985
- Van Riemsdijk, Henk and Edwin Williams. 1986. *An Introduction to the Theory of Grammar*. The MIT Press. Cambridge, MA
- Waite, William M. and Gerhard Goos. 1984. *Compiler Construction*. Springer-Verlag. New York
- Wehrli, Eric. 1984. "A Government-binding Parser for French." Institut pour les Etudes Sémantiques et Cognitives, Université de Genève. Working Paper No. 48

Appendix A: The Chain Rule

i. General rule and condition

attribution:

```
NP.Chain ← if NP.empty = '-' then NP.node  
           else if NP.pronominal = '+'  
                 then NP.node  
           else if NP.anaphoric = '+'  
                 then NP.A-Chain  
           else NP.Ā-Chain
```

condition:

NP.Chain ≠ 0

ii. Productions

a. Start production

Z → C

attribution:

```
C.A-Chain ← 0  
C.Ā-Chain ← 0
```

b. COMP productions

C → COMP'

attribution:

COMP'.x ← C.x, for x = A-Chain, Ā-Chain

condition:

C.Ā-Chain = 0

C → NP COMP'

attribution:

NP.x ← C.x, for x = A-Chain, Ā-Chain

COMP'.A-Chain ← C.A-Chain

COMP'.Ā-Chain ← NP.Chain

condition:

NP.Wh = '+'

COMP' → COMP S

attribution:

S.x ← COMP'.x, for x = A-Chain, Ā-Chain

c. INFL productions

S → NP INFL'

attribution:

NP.x ← S.x, for x = A-Chain, Ā-Chain

INFL'.A-Chain ←

if NP.θ_S = 'nil'
 then NP.Chain else 0

INFL'.Ā-Chain ←

if NP.Chain = S.Ā-Chain
 then 0 else S.Ā-Chain

INFL' → INFL VP

attribution:

VP.x ← INFL'.x, for x = A-Chain,
A'-Chain

d. V productions

VP → V'

attribution:

V'.x ← VP.x, for x = A-Chain, A'-Chain

V' → V NP

attribution:

NP.x ← V'.x, for x = A-Chain, A'-Chain

V' → V C

attribution:

C.x ← V'.x, for x = A-Chain, A'-Chain

V' → V NP C

attribution:

NP.x ← V'.x, for x = A-Chain, A'-Chain

C.A-Chain ← 0

C.A'-Chain ←

if NP.Chain = V'.A'-Chain
then 0 else V'.A'-Chain

e. N productions

NP → (NP₂) N'

attribution:

NP₂.A-Chain ← 0

NP₂.A'-Chain ← 0

N' → N (PP) (C)

attribution:

PP.A-Chain ← 0

PP.A'-Chain ← 0

C.A-Chain ← 0

C.A'-Chain ← 0

Appendix B: Functional determination of NP

i. General Rules

attribution:

NP.pronominal ←
if NP.empty = '-' then N'.pronominal
else if NP.Governor = <0,'nil'> then '+'
else '-'

NP.anaphoric ←

if NP.empty = '-' then N'.anaphoric
else if NP.Wh_S = '+' then '-'
else if NP.Governor = <0,'nil'>
then '+'
else if NP.Cases = 'nil' then '+'
else '-'

ii. Productions

NP → ε

attribution:

NP.empty ← '+'

NP → (Spec) N'

attribution:

NP.empty ← '-'