

A PARLOG Implementation of Government-Binding Theory

Robert J. Kuhns
205 Walnut Street
Brookline, MA 02146
USA

ABSTRACT

The purpose of this paper is to report on research on a parallel parser based on the principles and constraints of Government-Binding Theory. The parser outputs a set of licensing relations, notably thematic or θ -role assignments and antecedent/anaphor and other binding relationships. The primary goal of the system is as a linguistics tool for exploring concurrency and autonomy among modules of the theory.

BACKGROUND

Parsers based on Government-Binding (GB) Theory have been the subject of a number of research programmes.¹ While they all share a common linguistic framework, the design goals and underlying parsing strategies are varied. Some are tools for evaluating competence and/or performance models of language and others remain neutral with respect to psychological reality. The parsing engines range from bottom-up to top-down to mix-mode methods and from backtracking to deterministic algorithms. However, they all approach parsing as a sequential process, and although there have been a number of efforts on parallel parsing (cf. /Huang and Guthrie 1986/, /Matsumoto 1986/, and /Waltz and Pollack 1985/), few are linguistically-oriented and none are GB-based. Since GB is a highly modular system of components, development of parallel parsers incorporating the theory allows for a formal examination into the computational aspects of GB-modules and their interfaces.

SCOPE AND OBJECTIVES

In order to explore the relationships between GB-subsystems and their realization in a parallel parser, a development goal is one of modularity in that each GB-module should be transparently encoded.² This permits investigation of processes within and across GB components to be made explicit.

Since the focus of the parser is on GB principles and since GB itself is a theory of core grammar, the coverage of the parser is restricted to a subset of English that reflects syntactic processes explained by the theory.

While it may be premature to speak of psychologically-real parallel parsers, many of the cognitive presuppositions that are the basis of the work of /Berwick and Weinberg 1984/, /Marcus 1980/, and /Milne 1983/ have been the motivation for several of the design decisions that have been incorporated in the system. More specifically, the parser is deterministic and has the capability of delaying certain actions, e.g., projection of nodes and resolution of lexical ambiguity, until more information is available, rather than computing competing parses or backtracking uncontrollably. These features together with the goal of transparent representation of GB principles should provide a founda-

tion for future research into the cognitive plausibility of parallel parsers.

AN IMPLEMENTATION NOTE

The parser is being developed in PC-PARLOG³ requiring an IBM PC, XT, AT or compatible machine with at least 512Kb of memory and two floppy disk drives or a hard disk.

Although the implementation simulates parallelism by a timesharing scheduler, the parser runs on a single processor machine and, therefore, lacks true parallelism. However, insofar as the purpose of the parser is to specify concurrency of GB-subsystems at a high level of abstraction, the analyses are in terms of computational processes rather than processors. As parallel hardware becomes more readily available, a valuable by-product of this research is that the system could be ported to parallel logic machines with little or no additional effort.

PARLOG

PARLOG ("Parallel Logic") is a parallel declarative programming language based on logic (/Conlon 1989/, /Gregory 1987/, and /Shapiro 1988/). The basic form of a clause is

(1) head \leftarrow guard: body

where the **head** is a goal $r(t_1, \dots, t_n)$ and both **guard** (optional) and **body** (obligatory) are conjunctions of goals. Each procedure must be preceded by a mode declaration specifying input (?) and output (^) arguments. Sequential and parallel conjunction (AND) and sequential and parallel disjunction (OR) are represented by "&" and ":", ":", ":", ":", respectively.

THE PARSER

The parser combines top-down and bottom-up strategies and recovers a set of licensing relations directly; phrase structure is considered derivative and is not computed.⁴ The basic operations rely on the current governing category as the left bounded context and are deterministic in the sense that once a node is typed or licensed it cannot be altered. The output of the parser is a constructed list of structures

(2) CP = [cp(Complementizer).ext_arg(Subject),
infl(Infl_element),predicate(Verb),
int_arg_1(Object_1),int_arg_2(Object_2),
indirect_arg(PP)],

where **cp** represents the complementizer phrase, the head of the CP or S-bar, **infl** denotes the inflectional element, the head of S, and **predicate** is the verb. The structures **ext_arg**, **int_arg_1**, and **int_arg_2** are the arguments of the verb, the first being the

external argument (subject) and the last two being the (direct) internal arguments (objects). The **indirect_arg** is a prepositional phrase (PP) that a verb may license and there may be several of these structures depending on the lexical specifications of the verb. While the **cp**, **ext_arg**, **infl**, and **predicate** are obligatory, the internal and indirect arguments are contingent upon the θ -grid of the verb and, therefore, are optional.

The basic objects of the system are nodes where a node is a structure of the form **xp(Word, x_bar(Features), Index)**. **Word** is a lexical item, an empty category PRO, trace, or variable, or an empty complementizer or inflectional element, and **Features** represents the type of the node in terms of x-bar features $\pm N, \pm V$. Every node receives a unique index unless it is bound (co-indexed) to another via Binding or Control Theory. Each of the terms of (2) are nodes except for the arguments which are a list of nodes that represent the specifier, head, and complement structures. Specifiers may be determiner or adjectival phrases, and complements may be PPs or, in the case of relative clauses, a CP, which would be reflected in another list whose structure is analogous to (2).

THE LEXICON

In accordance with GB Theory, the lexicon plays a central role. Each lexical item contains the idiosyncratic features of the lexeme and they direct many actions of the parser. The lexicon is a database of PARLOG assertions which may be searched in parallel. A small subset of the lexicon is

- ```
(3) mode lexeme(item?, x_bar^, feature^).
 a. lexeme(man, [x_bar(n1, vo)], [pl(men)]).
 b. lexeme(put, [x_bar(no, v1)], [..., s(puts),
 ing(putting), θ -grid(int_arg_1,
 locative_PP), ...]).
 c. lexeme(plan, [x_bar(no, v1), x_bar(n1, vo)],
 [..., θ -grid (proposition),
 tenseless, subject_control, ...]).
```

The first term of each lexeme is the lexical entry and the second term is a list of structures of the form  $x\_bar(F1, F2)$ , where F1 and F2 are x-bar primitives,  $\pm N, \pm V$ , with "1" and "0" denoting "+" and "-", respectively. Lexically ambiguous items, e.g., (3)c., have more than one x-bar structure. The third term is a list of symbols and structures that depend on the lexical item and its type, i.e., x-bar features. For instance, as (3)a. illustrates, nouns ( $x\_bar(n1, vo)$ ) have a plural form as a member in their list of features while verbs ( $x\_bar(no, v1)$ ) have morphological derivatives, e.g., participial forms or **ing**, in their features list as (3)b. shows. Verbs also have  $\theta$ -grids which characterize argument structures and other features indicating specifics of complement structure such as tenseless (infinitival) subcategorizations or control criteria, e.g., **subject\_control** ((3)c.).<sup>5</sup>

In the current version of the parser, derivational morphology is minimal.

### THE PARSING ENGINE

The basic actions of the parser are to identify gaps, to construct maximal projections, and to license these projections, and these are encoded by the respective procedures, **detect\_gap**, **project**, and **license**. The parsing loop is called recursively until the sentential input, a list, is exhausted. During a parse, governing categories are produced and used as left-bounded context for certain procedures before being shunted to the list that will ultimately represent the output.

### GB MODULES

The GB-modules of Trace, Binding, Control, and Bounding Theories are incorporated in the parser, especially in goals **detect\_gap** and **license**. **Detect\_gap** identifies the presence, if

any, of an implicit element. This may be an argument, viz., trace, PRO, or variable, or a non-lexical item, viz., an empty complementizer or inflectional element. A fragment<sup>6</sup> of the PARLOG code for **detect\_gap** is

- ```
(4) mode detect_gap(sentence?, gov_cat^, empty_cat^).
    a. detect_gap([Word|Words], Gov_cat, Empty_cat) ←
        passive_morphology(Word, Gov_cat):
            Empty_cat = trace.
    b. detect_gap([Word|Words], Gov_cat, Empty_cat) ←
        Word = to, subjectless_inf_comp(Gov_cat):
            Empty_cat = pro.
```

The first relation has a guard to determine if the current word has passive morphology. Informally, the guard **passive_morphology** checks the inflectional element of the current governing category for a form of BE and, in parallel, determines if the current word is a verb of the passive form. If the guard succeeds, then the output substitution is complete with **Empty_cat** being unified with **trace**. In (4)b., the guards provide checks with the current token (**Word**) and governing category for the features of the verb and possible presence of an overt subject in order that **detect_gap** may detect a PRO. The rest of the specifications for **detect_gap** have similar strategies for finding variables and empty inflections and complementizers. Each of these clauses are computed in parallel together with one that determines an absence of an empty category in the current position of the sentence.

The next goal in parse is **project**, which constructs a maximal projection. **xp(Item, Type)**. In the case that **Item** is lexically ambiguous, **project** has calls to lexical disambiguation routines which are invoked in parallel to attempt a resolution.

The goal **license** indexes and assigns θ -roles to arguments of predicates. Binding Theory (/Chomsky 1981/) has three principles, frequently labeled A, B, and C in the GB literature, that specify co-indexation procedures for anaphors, pronominals, and referential (R-) expressions, respectively. Since the principles are independent, they are candidates for parallel execution. Thus, Binding is specified in the system as

- ```
(5) mode binding(proj?, gov_cat?, indexed_proj^).
 binding (Proj, Gov_cat, Indexed_proj) ←
 is_anaphor(Proj):
 principle_a(Proj, Gov_cat, Indexed_proj).
 binding(Proj, Gov_cat, Indexed_proj) ←
 is_pronominal(Proj):
 principle_b(Proj, Gov_cat, Indexed_proj).
 binding(Proj, Gov_cat, Indexed_proj) ←
 is_r_exp(Proj):
 principle_c(Proj, Gov_cat, Indexed_proj).
```

The purpose of the guards is to determine the argument type of a particular argument (**Proj**) which then invokes a call to a specific method of indexing in accordance with Binding Theory. (There is a numerical indexing scheme that is embedded in each principle.) The result of **binding** is a node, **Indexed\_proj**, that has an index associated with it.

The other major part of **license** is the assignment of  $\theta$ -roles. The parser assigns  $\theta$ -roles to the respective arguments of the predicates based on their  $\theta$ -grids. The binding and  $\theta$ -assignment procedures together comprise the licensing procedure. Their communication channels are being explored and stream AND parallelism appears promising.

## AN EXAMPLE

For the sentence

(6) John was persuaded to leave.

the parser produces

```
(7) [cp(emp,cp,1),
 ext_arg(john,x_bar(n1,vo),2),
 infl(was,infl,3),
 predicate(persuaded,x_bar(no,v1),4),
 int_arg_1(trace,x_bar(n1,vo),2),
 int_arg_2(cp_1,cp,5),
 [cp_1(emp,cp,5),
 ext_arg(pro,x_bar(n1,vo),2),
 infl(to,infl,6),
 predicate(leave,x_bar(no,v1),7)]]
```

which illustrates PRO and trace detection and binding.

## FUTURE DIRECTIONS

The primary focus of the experimental parser will be to include wider coverage within a GB-framework, including adjuncts, and to examine psychological aspects of concurrent language processing.

## CONCLUSION

The investigation into the concurrent aspects of GB Theory is still in the embryonic stages. Nevertheless, early results seem to indicate that such research may yield significant insights into parallel parsing and human sentence processing.

## NOTES

<sup>1</sup>Cf. /Abney 1986/, /Berwick 1987/, /Correa 1987/, /Kashket 1987/, /Kuhns 1986/, /Sharp 1985/, /Stabler 1987/, /Thiersch 1988/, and /Wehrli 1984/.

<sup>2</sup>/Berwick and Weinberg 1981/ and /Wehrli 1988/ have excellent discussions of the notion of transparency or direct encoding of GB-modules.

<sup>3</sup>PC-PARLOG is commercially available from Parallel Logic Programming Limited (Twickenham, England).

<sup>4</sup>/Berwick 1987/ presents various output options with respect to GB-based parsers. /Abney 1986/ also has a novel approach to parsing.

<sup>5</sup> $\theta$ -roles are denoted by the neutral labels `ext_arg`, `int_arg_1`, and `int_arg_2` rather than by the terms of agent, theme, recipient, etc. /Zubizarreta 1987/, for example, argues for this type of representation.

<sup>6</sup>For the sake of expository simplicity and for lack of space, low-level implementation details, e.g., certain initialization and stack manipulation procedures, are suppressed in the discussion.

## REFERENCES

- Abney, S., (1986) "Licensing and Parsing," (personal communication).
- Berwick, R.C., (1987), Principle-Based Parsing, Technical Report 972, MIT Artificial Intelligence Laboratory, Cambridge, MA.
- 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.

Conlon, T., (1989), Programming in PARLOG, Addison-Wesley, Reading, MA.

Correa, N., (1987), "An Attribute-Grammar Implementation of Government-Binding Theory," Proceedings of 25th Annual Meeting of the ACL, Stanford, CA.

Gregory, S., (1987), Parallel Logic Programming in PARLOG, Addison-Wesley, Reading, MA.

Huang, X., and L. Guthrie, (1986), "Parsing in Parallel," Proceedings of COLING '86, Bonn, West Germany.

Kashket, M.B., (1987), A Government-Binding Based Parser for Warlpiri, Technical Report 993, MIT Artificial Intelligence Laboratory, Cambridge, MA.

Kuhns, R.J., (1986), "A PROLOG Implementation of Government-Binding Theory," Proceedings of COLING '86, Bonn, West Germany.

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

Matsumoto, Y., (1986), "A Parallel Parsing System for Natural Language Analysis," Springer Lecture Notes in Computer Science, no. 225.

Milne, R., (1983), Resolving Lexical Ambiguity in a Deterministic Parser, D. Phil Dissertation, University of Edinburgh.

Reyle, U., and C. Rohrer, (eds.), (1988), Natural Language Parsing and Linguistic Theories, D. Reidel Publishing Co., Boston, MA.

Shapiro, E., (ed.), (1988), Concurrent Prolog: Collected Papers, The MIT Press, Cambridge, MA.

Sharp, R., (1985), A Model of Grammar Based on Principles of Government and Binding, M.S. Thesis, The University of British Columbia.

Stabler, E.P., (1987), "Restricting Logic Grammars with Government-Binding Theory," Computational Linguistics, vol. 13, no. 1-2.

Thiersch, C., and H.P. Kolb, (1987), "Parsing with Principles and Parameters: Prolegomena to a Universal Parser," (personal communication).

Waltz, D.L., and J.B. Jordan, (1985), "Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation," Cognitive Science vol. 4, no. 1.

Wehrli, E., (1984), "A Government-Binding Parser for French," Working Paper No. 48, Institut pour les Etudes Semantiques et Cognitives, Universite de Geneve.

Wehrli, E., (1988), "Parsing with a GB-Grammar," in Reyle and Rohrer (eds.).

Zubizarreta, M.L., (1987), Levels of Representation in the Lexicon and in the Syntax, Foris Publications, Dordrecht, Holland.