North East Linguistics Society

Volume 16
Issue 1 *NELS 16*

Article 2

1985

A Government-Binding Parser

Steven Abney
Massachusetts Institute of Technology

Jennifer Cole
Massachusetts Institute of Technology

A Government-Binding Parser

Steven Abney
Jennifer Cole
Massachusetts Institute of Technology

Introduction

Grammatical theory gives an account of knowledge of language. In the generative enterprise, an understanding of grammar has been taken to be prior to the exploration of language processing, at least methodologically. It has frequently been emphasized in the generative literature that no theory of linguistic behavior can attain any degree of explanatory adequacy if it does not incorporate an adequate theory of linguistic knowledge. What has received considerably less attention, though, is the complementary truth: that the intent of a theory of linguistic knowledge is to form the core of a theory of language processing. A theory of grammar would be of little interest if it were manifestly incapable of supporting reasonable models of linguistic behavior.

But the advances of recent years in grammatical theory have had relatively little influence on the study of language processing. This is understandable, considering that contemporary generative theory has developed with little attention paid to processing, and hence presents no obvious procedural interpretation for its constructions. In this paper, we present a particular procedural interpretation of Government-Binding theory—that is, a model of parsing—as a contribution to a more general theory of linguistic processing which incorporates a principle-based theory of grammar.

We see the task of designing a principle-based parser as consisting in translating grammatical principles into procedures, which perform the task of parsing. The principles of grammar are well-formedness conditions on syntactic structure. As such, it would be trivial to translate them into procedures which check structures for well-formedness, but it is rather more difficult translating them into procedures which build structures in conformity with those principles. We solve this problem by capitalizing on the idea that the primary condition on syntactic structure is that each node be *licensed*, and that most other conditions on structure are ultimately subordinate to the licensing conditions. Structure-building is driven by deciding how each incoming node is to be licensed in the developing structure, and working out the implications of that decision.

1 Background

1.1 Licensing

We assume that the primary condition on syntactic structure is that each node be licensed. We may suppose that non-maximal categories are licensed by heading maximal categories, in conformance with $\overline{\text{X}}$-theory. The distribution of maximal categories is not sufficiently constrained by $\overline{\text{X}}$-theory, however, and we assume that each maximal category is licensed by entering into a sufficiently strong relation with an independently-licensed node. The "sufficiently strong" relations include θ-assignment, predication, and functional-selection (see below).

By "θ-assignment" we mean only *direct* θ-assignment. We assume that subjects of sentences, whether θ-assigned or not, are licensed by predication. In this way, we may treat θ-assignment (in English) as uniformly right-directional, and predication as uniformly left-directional.

We take predication to license subjects, secondary predicates, and modifiers. This is a rather diverse class of elements, and it may be necessary to divide up the burden of licensing them, though we will not pursue the question here. One point of note is that we assume *all* subjects to be licensed by predication, including the subject of CP (i.e., $\overline{\text{S}}$), even though there is no semantic relation between the "predicate" (namely, IP (S)) and the "subject" (namely, a fronted *wh*-element). This is consistent with Rothstein's (1983) suggestion that the Extended Projection Principle (Chomsky (1981)) be derived by supposing that even pleonastic subjects of sentences are predicated of, but it does involve treating predication as a "purely" syntactic relation.[1]

We assume that the heads of S and $\bar{S}$ are I(nfl) and C(omp), respectively, following Chomsky (forthcoming). Following Abney (1985), we assume that NP's have a similar structure: that the head of NP is actually the determiner (D). I, C, D, and (possibly) P constitute the class of *functional elements*. Functional elements lack intrinsic referentiality (in a sense that generalizes over both nominal and verbal referentiality: roughly, semantic predicatehood) and inherit their referential properties from their complements. The relation between a functional element and its complement is *functional selection*. VP, IP, etc., are licensed by functional selection. Functional selection is right-directional, and obtains only under adjacency.

The assumption that Determiners are also functional elements allows us to parse noun phrases (i.e., DPs) on a principled basis, for a large part. It is likely that certain aspects of noun phrase structure will be fixed by language-specific rules, but it is our intent to account for as much as possible through universal principles.

Licensing-relations are used to recover a skeletal syntactic structure, which is refined by other components of grammar such as binding and movement. There are conditions on syntactic structure other than those imposed by the licensing requirements. For instance, θ-theory not only licenses arguments, but also requires that every (obligatory) θ-role be assigned. Binding theory does not contribute to building the phrase structure tree, but defines coreference possibilities between elements already in the tree. In general, these additional requirements on structure are responsible for inserting empty elements, and adding relations over and above licensing and phrase-structure relations. We claim that they have a very limited effect on the building of structure, however, and this claim seems to be true over a broad range. These issues will be taken up as we discuss the various components of grammar.

1.2 The computational model

The model of computation we adopt is the Actor model being developed in the Apiary project at the MIT AI Laboratory. For the purposes of this paper, the details of the actor model are not important. We present here a very brief overview of the model; for more information, the reader is directed to Agha (1985).

In the actor model, calculation is distributed among various computational agents, or *actors*. Actors all operate in parallel; each can be imagined as a separate microprocessor. They coordinate their actions by means of message passing. Each actor has a number of *acquaintances*, or other actors whose *mail* *address* it knows; these are the actors to which it can send messages. Upon receiving a message, an actor can take any number of three kinds of action: it may create other actors, it may send messages to other actors, and it may change its behavior relative to the processing of future messages.

One implementation detail that is of some significance is the use of *futures*. Suppose that some actor asks a node n for an acquaintance α of n's, but the identity of α is not yet known. Take for example, the case where n is a certain word, and α is the following word—n's NEXT acquaintance—which has not yet been input. An actor requesting the identity of α should not have to make allowances for such a situation. We would like n simply to wait until α's identity is known, before replying to the caller. At the same time, it is messy and inefficient for nodes to juggle long lists of callers and requests, waiting for information to become available. The device we use in this case is to represent unknown acquaintances with *futures*. A future is a special actor that accepts all incoming messages, and buffers them until the identity of the actor it is standing in for—its *replacement*—is known. When the future's replacement is determined, it passes the requests it had buffered to its replacement for handling. Thus there may be a very long wait between sending a future a request, and receiving the reply, but the possibility of such a wait is inherent in the message passing system, even apart from futures.

The use of futures effectively allows us to abstract away from the fact that the sentence is being input left-to-right. We can design algorithms which function regardless of whether the information they require lies to the left or to the right. This is important if we wish to use the same algorithms to parse languages that differ in the settings of directionality parameters: e.g., left-headed languages like English vs. right-headed languages like Japanese. At the same time, the number of processes which are waiting on right context at any point in the parse gives a rough measure of the psychological complexity of the parse.

2 The Parsing Model

2.1 The representation recovered

The parser takes a string of words as input, and generates a syntactic structure for that string. A syntactic structure consists of a set of nodes, and the linguistic relations defined on those nodes. Each node is represented by an actor, and the relations in which that node participates are represented by the node's acquaintances.

Computation is performed by the various components in parallel, as new words come in, or as new nodes are created. Each component has one actor whose responsibility is to accept new incoming nodes and decide if its component has an interest in them, and if so, which actor or actors in its component would be interested in them. Nodes incoming to binding theory, for instance, would be of interest only if they are arguments; and then, they would be of interest to different actors, depending on whether they are anaphors, pronouns, or R-expressions.

Let us go through a sample parse, to get the flavor of the model. In succeeding sections we will go through the various components more carefully.

Consider a very simple sentence, such as

John threw the ball

We assume a front-end morphological analyzer that recognizes Infl in the verbal morphology and "undoes" the effects of Affix-hopping, yielding

John PAST throw the ball

$\overline{X}$-theory projects each word to the $\overline{X}$ and X^{max} levels. As soon as *John* has been projected to NP,[2] the NP node begins searching for a licenser. NP approaches Infl as a potential licenser, but Infl functionally selects VP, and is incapable of licensing NP by either predication or θ-assignment. NP approaches V. V has a θ-role available, but it cannot assign that role to NP because the θ-directionality parameter is set for rightward assignment in English. Finally, NP finds that VP is able to license it by predication.

In the meantime, Infl seeks a complement from which to receive a semantic value. This complement must be a VP, and must be right-adjacent; fortunately, there is a VP that fits the bill. The licensing relations for NP and VP having thus been established, government theory steps in to calculate precisely where the NP and VP must be attached.

V begins seeking for an argument to receive its obligatory θ-role. When the object NP, *the ball*, is constructed, it is immediately taken up by the V as the desired argument. Again, government theory steps in to determine precisely which node is the object's parent.

2.2 $\overline{X}$-theory and government

In assembling the phrase structure tree, $\overline{X}$-theory and Government are ubiquitous. $\overline{X}$-theory's most important role is in creating the $\overline{X}$-projections of

North East Linguistics Society, Vol. 16 [1985], Iss. 1, Art. 2

incoming words. An $\overline{\text{X}}$ actor creates the projections of the word, assigns them the proper bar-levels, and establishes the dominance and headship relations between them.

Government serves as an intermediary between licensing relations and $\overline{\text{X}}$-relations. The key decisions in building structure are the decisions concerning how each node is to be licensed. Once a licensing relation is established for a node, the matter can be turned over to government theory, which, in conjunction with $\overline{\text{X}}$-theory, determines precisely where the node should be attached.

When government is called, it is given the identity of a governor, whose attachment in the tree is assumed to be known or forthcoming, and a governee, whose attachment is to be determined. Define the *domain* of a node to be the first maximal category properly dominating that node. The government actor finds the domain of the governor, and attaches the governee under the domain, or under its immediate head. If both positions are possible attachments, the lower position is preferred: in general, a 0-level node imposes stronger restrictions on its sisters than do 1-level nodes (in fact, intermediate-level nodes apparently never place any restrictions on their sisters); it is assumed that nodes occupy the most restrictive position they are capable of occupying.

This constructive algorithm has an even simpler testing counterpart: to test if two nodes govern one another, it is only necessary to check that they have the same domain.

2.3 θ-theory

For arguments, there are two possible licensing relations: θ-assignment and predication. Roughly, θ-theory licenses complements, and predication licenses subjects (as well as adjuncts, and possibly modifiers). We take up predication in the next section.

θ-theory is also responsible for seeing that all obligatory θ-roles are assigned. If one is not, θ-theory inserts an empty category, which must then satisfy independent constraints on empty categories.

When a θ-assigner is input to θ-theory, θ-theory recognizes that it has a θ-grid, and assigns it an actor which is an "expert" at getting θ-grids satisfied. If a given θ-role is obligatory, θ-theory actively seeks a receiver for it. Otherwise, it is assigned a receiver only if the verb is approached by an argument seeking to be licensed.

Certain θ-roles are annotated with a prepositional class. When an attempt is made to assign such a θ-role, the argument must be a prepositional phrase whose head belongs to the class in question.

There are cases of ambiguity which must be dealt with. Consider the sentence:

I recited a sonnet to a countess

Suppose *I recited a sonnet* has been analyzed. and it is the attachment of the PP *to a countess* that is in question. The PP could be licensed by either the N *sonnet* or the V *recited.* It has been noted that in such cases construal with the verb is much preferred (e.g. Kimball (1973), Ford, Bresnan, and Kaplan (1982)). The preference is strong enough to make it difficult to find the plausible reading of the following sentence:

Hang the sign on the elephant on the flagpole

The fact that the "correct" reading of this sentence is difficult to find is evidence against an approach like that of Marcus et al. (1982), in which a representation is developed (called *D-theory*) which will allow the parser to put off making a definitive decision concerning the attachment of such PP's until all potentially relevant information has been collected. Apparently the parser only waits until the PP is complete before making an attachment decision, and if that decision turns out to have been ill-advised, processing difficulty results.

These facts are accounted for in the present model in the way arguments seek licensers. Roughly, the decision procedure is this: an argument approaches the verb first, then other potential θ-assigners to its left. If no suitable assigner is found. it begins seeking to its right.

The decision procedure limits the number of potential assigners which an argument considers. This introduces the possibility of sentences to which the grammar assigns a structure, but whose structure the parser is incapable of recovering. precisely because of the limitations imposed by the decision procedures used by arguments. An example is the following:

I put the ball that Bill threw on the table

Given certain formulations of the decision procedure arguments use when seeking licensers. the PP *on the table* would fail to find the potential licenser *put.* and choose instead *threw.* On the basis of this decision, the parser would judge the sentence ungrammatical (because *put* fails to find a receiver for an obligatory θ-role, and complains), even though it is assigned a well-formed structure by the grammar. In fact, humans make the same error parsing this sentence the first time, and the restrictedness of arguments' decision procedure gives an account of this fact.[3]

In a broader view, what this points out is that there are two sets of constraints imposed on the parser: grammatical constraints, and "psychological" or "performance" constraints. We are primarily concerned with grammatical constraints in the present paper, but we do not wish that to be construed as a lack of appreciation for performance constraints. A complete model must instantiate both sets of constraints, and though our primary concern at this stage has been the incorporation of grammatical constraints, we believe that additional constraints which provide an account of human performance limitations—such as the decision procedure sketched above—can readily be incorporated into our model.

2.4 Predication, functional selection

θ-theory handles licensing for an important class of cases, viz., the complements of verbs, nouns, and adjectives. There are a number of cases that remain to be accounted for, however. Predication licenses subjects and adjuncts (perhaps also modifiers); functional selection licenses VP and NP (distinguishing NP from DP now), and IP and DP, when they are the complements of C and P, respectively.

Primary predication governs the licensing of subjects by predicates. External θ-assignment is "parasitic" on predication. When a predication relation has been established between an argument and a predicate, θ-theory is called into play to establish a θ-assignment relation as well, if there is a θ-role to be assigned. If no θ-role is available, movement theory is called into play to establish a chain, by means of which the argument can receive an interpretation (see below, "Movement").

Functional selection is the relation between functional elements and their complements. Like θ-assignment, it works in two directions: not only do certain nodes require functional elements to be licensed, but functional elements also require complements in order to acquire referential properties. The relation between a functional element and its complement is even stricter than that between a θ-assigner and its complement: the functional element's complement must be right-adjacent. For this reason, functional elements generally know very quickly whether or not their complement is forthcoming, and can drop a trace or stop the parse early, if one is not forthcoming.

NP, VP, and (in most cases) IP, require a functional element as licenser, and cannot be licensed by θ-theory or predication. This is apparently the reason why their functional elements (D, I, and C, respectively) can frequently be empty. Finding a NP or VP without a D or I provides enough information to know to drop an empty functional element immediately to the left.

2.5 Binding

The role of binding theory is to check, for every NP,[4] that the incorporation of that NP into the phrase structure tree does not violate any of the principles of binding theory. Binding functions for the most part in checking syntactic structures, rather than building structure. For anaphors, binding theory identifies all NPs that are in the binding domain of the anaphor and could serve as *potential* antecedents.[5] This list must be non-null for the sentence to be well-formed. We assume that a semantic component, not belonging to the parser proper, is responsible for binding the anaphor to one of its potential antecedents. In the case of pronouns and R-expressions, binding theory identifies all "anti-binders": those NPs in the binding domain which c-command the pronoun or R-expression. The semantic component uses the list of anti-binders in assigning indices to the pronouns and R-expressions; the pronoun or R-expression cannot be coreferential with any anti-binder. The list of anti-binders is very much like the anaphoric indices of Chomsky (1980).

2.6 Movement

The role of movement is to provide operators and non-θ-marked arguments with an interpretation. When an operator or argument attaches itself in a non-θ-marked position, movement theory creates a *chain* actor which searches for a position for an empty category. Subjacency checks chains formed this way, but is not built into the method by which chains seek gaps. Thus the parser can build structures for sentences which violate subjacency, even though it recognizes them as ill-formed.

Both A- and $\bar{A}$-chains are represented by chain actors, which are created upon identification of either an operator or a non-θ-marked argument. The first member of the chain will be the operator or argument that triggered its formation. Subsequent links are added to the chain in accordance with the principles of Chain Formation, as presented in Chomsky (1986).

A chain is *active* until it acquires a final link which occupies either an argument or adjunct position (the latter being possible only in the case of $\bar{A}$-chains). A parse is said to fail if there are any active chains at completion.

Let us consider how the empty category that fills the final link of a chain is created. Any *ec* must meet the same licensing requirement that holds of every node in the phrase structure tree; it must find a licenser. θ-assigners license *ec*'s in argument positions. Those *ec*'s that are in adjunct positions must be licensed via predication. Therefore, a natural place for active chains to search for their final links is with the licensers in the sentence.

Active chains scan on incoming words, and whenever a licenser is identified, the chain asks it if it can license an empty category. Licensers will postulate empty categories only if they have roles to assign which cannot be filled by overt arguments in their licensing domain. If a licenser has an obligatory role to assign, it will postulate an *ec* even before being approached by a chain actor. If a licenser has postulated or can postulate an *ec* which meets the chain's requirements, it responds to the inquiring chain with the address of that *ec*. If all licensers have been queried and no licensed *ec* has been created at the end of the chain's c-command domain, then the chain will complain and stop the parse.

If a chain actor does not find an *ec* within a certain bounded domain, it posits an intermediary trace in Comp to serve as the next link in the chain.[6] If a chain actor does not encounter either a final link within the bounded domain, or a position in which to posit an intermediate trace, then it may extend the domain of its search, but it marks the sentence as a mild subjacency violation. If the search fails in the extended domain, the sentence is marked ungrammatical and the parse is stopped. (Alternatively, domain extensions can be continued indefinitely, but the sentence is marked *n*-subjacent, for *n* the number of domain extensions.)

We can see that the subjacency constraint is imposed on chain formation, but chain formation is not itself defined in terms of subjacency. In this way, the parser will succeed in associating antecedents and gaps, even in structures that violate subjacency, without having to introduce special interpretive conventions.

3 A Broader Perspective

3.1 Principle- versus rule-based parsers

Principle-based parsing models are conspicuously absent in the psycholinguistic and computational literatures. The majority of parsing models assume an augmented context-free grammatical theory, and some version of one of the standard CFG parsing algorithms.[7] The popularity of the CFG paradigm is understandable. Context-free grammars enjoy a long tradition (relatively speaking), and they are mathematically well-understood. They are adequate—or very nearly adequate—to perform the task of a grammar, in the classical definition of that task: to generate all and only the sentences of the language, and to assign to each sentence its proper structure(s). Even linguists who consider the phrase structure representation generated by context-free grammars to be inadequate, nevertheless consider phrase structure to be an important aspect of syntactic structure. Finally, the context-free formalism is simple and mathematically appealing, yet permits grammars of only restricted generative capacity, and for which there exist provably efficient parsing algorithms: i.e., algorithms which parse in $O(n^3)$ time, for n the length of the sentence.[8]

CFG-based parsing models have serious deficiencies, however. Context-free rules generate representations which include only the configurational relations, dominance and precedence. There are many extra-configurational relations, though, which are linguistically significant (e.g., "long-distance" relations such as binding and movement); and there are also local relations, such as θ-assignment, which cannot be defined in strictly configurational terms.

In a strictly phrase-structure representation, extra-configurational relations can be expressed only by means of special devices, such as the slash categories of Generalized Phrase Structure Grammar (GPSG) or the metavariables (the up- and down-arrows) of Lexical-Functional Grammar (LFG).[9] Alternatively, such relations are consigned to a "semantic" component. But the advantages of the context-free paradigm, including the parsing complexity results, of course do not extend to calculations done on semantic structures. An example is the functional structures of LFG. It should be noted that most of the criticisms leveled here against context-free systems—in particular, those concerning acquisition—do not apply to LFG as a whole, because of the contribution of functional structure. But the addition of functional structure makes LFG parsing, in the worst case, NP-hard (see Berwick (1982)).

To whatever extent extra-configurational relations cannot be directly represented—and the more closely a strictly context-free formalism is adhered to, the greater that extent—they must be represented meta-grammatically, if at all. A valid question is then whether they need to be represented. At this point, then, we review the motivations for a theory in which extra-configurational relations are primary, and discuss several advantages a principle-based theory has over rule-based theories in accounting for human language processing.

3.2 Language Acquisition in Principle Theories

One of the fundamental reasons why a broad range of diverse principles are adopted as the objects of description in LGB and related work is in order to explain language acquisition. Research in generative grammar has shown that not only is language extremely complex, but speakers have clear and consistent judgements for most grammatical structures. Moreover, it has been determined that many aspects of language acquisition occur on the basis of insufficient or no evidence.

The model which has emerged to account for the fact of language acquisition under these circumstances is one in which the "core" of linguistic knowledge is not learned, but rather is innate. The same set of grammatical principles apply in every language, modulo limited parameterization. However, the diversity of the surface forms of language forces one to state these principles at a considerable degree of abstraction. Context-free systems, on the other hand, by emphasizing the configurational aspects of language, emphasize an aspect of language which varies greatly from language to language. There are no context-free rules which are universal. CFG-based theories are thus faced with a much larger task in accounting for acquisition, as much more must be acquired.

In the GB framework, the process of language acquisition can be viewed as a process of enhancing an abstract representation of a principle-based grammar. When a child encounters a sentence that is not accounted for by the simple representation of grammar he has so far acquired, he is able to isolate precisely which aspects of his grammar are insufficient, in terms of principles. Since the representation of grammar is maximally transparent to the statement of grammatical principles, it is possible for the learner to make any appropriate adjustments to accomodate the new sentence.

Consider what would happen if we were to adopt a context-free formalism for the principle-based grammar which the parser accesses. Such a design would necessitate the existence of a device for "compiling" a modified abstract grammar into a new "concrete" (i.e., context-free) grammar for the parser. In general, very small changes in the abstract grammar may have wide-ranging ramifications for the concrete grammar, and may necessitate the modification of an arbitrary number of concrete-grammar rules. Given a grammar in this format, it is clear that the task for the acquisition device would be enormous. In order for the parser to localize a failure with respect to principles of the grammar, it would first need to "uncompile" the phrase-structure rules of the grammar to determine which principle(s) is at fault. The design of either a compiler for GB, or its reverse function, would be a formidable, if not impossible, task. To our knowledge, it has never been attempted.

In short, it appears that the adoption of a CFG-based grammatical theory buys ease of parsing at the expense of an explanation for acquisition.

3.3 The Analysis of Ungrammaticality

In addition to providing an account of language acquisition, a principle-based theory of grammar also sheds light on the ability of humans to interpret and discriminate between ungrammatical utterances. To illustrate, we contrast the behavior of rule-based and principle-based parsers in analyzing ungrammaticality.

In rule-based systems, parsing proceeds by matching items from the input string against rules in the grammar. When an item is encountered for which there is no corresponding phrase structure rule that is consistent with the existing structure, the parser fails and the sentence is marked ungrammatical. The only information that the parser has about the nature of the failed parse is the structure that was assigned up until the point of failure, and the identity of the item on which the parse failed.

The existence of interpretable but ungrammatical sentences poses a problem for rule-based parsers. For some cases of ungrammaticality, like subjacency violations, humans are able to interpret the ungrammatical sentence; this implies that the human parser assigns structure to some ungrammatical sentences. Inasmuch as rule-based parsers are unable to assign structure to ungrammatical sentences, they fail to reflect the behavior of humans. Moreover, the human parser is also able to consistently define differences in degree of ungrammaticality, such as that between subjacency and ECP violations. A rule-based parser cannot distinguish degrees of ungrammaticality because it does not know *why* a sentence is ungrammatical.

A principle-based theory of grammar like GB explains ungrammaticality as the violation of one or more of the principles of grammar. When a principle-based parser encounters an ungrammatical sentence. an actor representing a principle of grammar complains about some aspect of the structure. However, it is not necessary that all syntactic analysis halt whenever a principle is violated; a principle-based parser may be designed so as to allow structure-building to proceed. even in light of such violations. Given a modular grammar like GB, any aspect of the syntactic analysis that is independent of the component where a violation occurs may continue unfettered.

3.4 The Parser-Grammar Relationship

The model that emerges from the implementation described in earlier sections is one in which the parser and the grammar are no longer clearly discrete and autonomous entities. The grammar is intrinsically defined by the actions of the parser. This model implies that there is no unified representation of the grammar as a set of declarative statements. Rather. grammar is defined as an abstraction from the constrained procedures of the parser. We say abstraction because, in addition to grammatical constraints. the parser is also constrained by performance limitations. Performance constraints may affect the amount of work space the parser has. or the application of search algorithms, etc. However, grammatical principles are still the building blocks of the parser, since each procedure corresponds to some grammatical principle and the constraints it imposes on structure-building.

Having grammatical information so closely related to procedural information is feasible because the grammar being implemented is Universal Grammar. Grammatical information is encoded directly into the parsing mechanism and the entire device is part of our innate language faculty. We assume that parameters (such as word order) are identified as they are encountered in the acquisition process. and they are subsequently fed into the parse, which in turn can alter its behavior to reflect these parameter settings. Specifically. certain actors reference a list of parameter settings to decide how they should behave. The parsing mechanism remains constant across languages, varying only in pre-determined ways to accommodate a new lexicon and parameter settings.

Most models of rule-based parsing maintain a distinction between the parser and the gammar (perhaps the sole exception being ATN's). In these models the parser is seen as a general procedural device that can apply itself to any CF grammar, using the same procedures each time to build structure. In fact, a separate grammar exists for each language the parser analyzes; the set of possible grammars being constrained only by the meta-grammar which encodes certain grammatical constraints.

In implementing a GB grammar, it would not be possible to adhere to a design in which a generalized parser accesses various autonomous grammars. This is due to the fact that there is no metagrammar of GB theory constraining the statement of grammatical principles. There are no theoretically defined limits on the extent to which the information which a given principle regulates is dispersed over the tree. The implications of principles for phrase structure may be mediated by arbitrarily complex abstract constructs. The task of designing a parser that can take an arbitrary set of such principles and parse in accordance with them is clearly intractable. But GB theory is not intended as a theory in which each language has a complete and independent grammar; it is a theory of Universal Grammar. All the principles stated in the grammar are applicable to each natural language. Thus, there would be no advantage to designing a principle-based parser that kept the notion of the parser as a general procedural device, since it would necessarily be accessing only one grammar—the Universal Grammar.

3.5 A General Linguistic Processor

This brings us to a discussion of the relations between the parser and the generator and acquisition devices. In the past, researchers have advocated models that separate the parser from the grammar in order to be able to have both the parser and the generator access the same representation of the grammar. In this way they avoided redundantly specifying grammatical information. Since in our model of parsing, grammatical information is inextricably tied up with procedural information, the question arises: How do we avoid such redundancies? We propose that the actors that encode grammatical principles are defined so as to reflect those principles while performing the tasks of parsing, generation and language acquisition. Our concern in the present paper has been with describing a parser, but we envision a more general linguistic processor, where the tasks of parsing, generation, and acquisition are distinguished at the actor level, rather than being performed by distinct devices.

4 Conclusion

We have argued for the importance and feasability of a model of parsing which instantiates Government-Binding theory. This is important both as an enrichment of our understanding of GB theory, and as an enrichment of our understanding of natural language processing. We have proposed a parsing model in which the parser embodies the grammar (rather than merely referencing the grammar); this is feasible because of the universal nature of the grammar. The universality of the parser is also significant of itself. Parsing proceeds by choosing certain aspects of grammatical knowledge as primary for building structure, namely, the licensing relations.

Acknowledgements

We have benefitted from discussions with the following people: Ed Barton, Bob Berwick, Noam Chomsky, Janet Fodor, Kyle Johnson, Doug Saddy, and Carol Tenny. Our mentors on the computational side have been Gul Agha and Tom Reinhardt. Thanks to Carl Hewitt and the Apiary Project of the MIT AI Laboratory for providing the computational framework and resources.

Steven Abney's work is partially supported by a Mellon Fellowship in the Humanities. Jennifer Cole's work is partially supported by a grant from the National Science Foundation. The Apiary Project is supported by a grant from the System Development Foundation. The Artificial Intelligence Laboratory is supported in part by the Advanced Research Projects Agency of the Department of Defense.

Footnotes

[1]A possibility which we will not pursue would be licensing subjects via abstract spec-head "agreement" (see Chomsky (forthcoming)).

[2]Ignoring DP now for simplicity's sake.

[3]When people succeed in parsing the sentence, it is because "higher-level" heuristics step in to try to determine why the parser failed, and to cue it on a second pass.

[4]We return to using the traditional notation "NP" in this section. Technically, "DP" should be understood throughout.

[5]The definition of binding domain is given in Chomsky (1986).

[6]The "bounded domain" is defined in terms of *barriers*. The reader is referred to Chomsky (forthcoming). There it is also argued that intermediate traces appear not only in Comp, but also adjoined to VP.

[7]This is not to imply that parser and grammar are kept distinct in the implementation: an obvious exception is the ATN parsers.

[8]Though there is considerable doubt whether the computational sense of efficiency involved is linguistically relevant.

[9]Though it is not to be supposed that such devices are a new idea for extending the descriptive power of context-free grammars: a similar device was proposed by Chomsky as early as 1949, in an undergraduate thesis.

References

Abney, S. (1985) "Functional Elements and Licensing", ms., MIT.

Agha, G. (1985) *Actors: A Model of Concurrent Computation in Distributed Systems*, AI Technical Report 844, MIT AI Laboratory, Cambridge MA.

Berwick R. (1982) "Computational Complexity and Lexical-Functional Grammar", *American Journal of Computational Linguistics* 8.3-4.

Chomsky, N. (1980) "On Binding", *Linguistic Inquiry* 11.1

—— (1981) Lectures on Government and Binding Foris, Dordrecht

—— (1986) Knowledge of Language: Its Origins, Nature, and Use, Praeger Publishers, NY.

—— (forthcoming) "Barriers". Linguistic Inquiry Monograph.

Ford M., J. Bresnan. R. Kaplan (1982) "A Competence-based Theory of Syntactic Closure", in Bresnan, ed., The Mental Representations of Grammatical Relations, MIT Press, Cambridge MA.

Kimball, J. (1973) "Seven Principles of Surface Structure Parsing", *Cognition* 2.1.

Marcus M., D. Hindle, M. Fleck (1982) "D-Theory: Talking about trees", *Proceedings of ACL 1982.*

Rothstein. S. (1983) "The Syntactic Form of Predication", PhD diss., MIT.