MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 788 July, 1984

TOWARD A PRINCIPLE-BASED PARSER

G. Edward Barton, Jr.

ABSTRACT:

Parser design lags behind linguistic theory. While modern transformational grammar has largely abandoned complex, language-specific rule systems in favor of modular subsystems of principles and parameters, the rule systems that underlie existing natural-language parsers are still large, detailed, and complicated. The shift to modular theories in linguistics took place because of the scientific disadvantages of such rule systems. Those scientific ills translate into engineering maladies that make building natural-language systems difficult.

The cure for these problems should be the same in parser design as it was in linguistic theory. The shift to modular theories of syntax should be replicated in parsing practice; a parser should base its actions on interacting modules of principles and parameters rather than a complex, monolithic rule system. If it can be successfully carried out, the shift will make it easier to build natural-language systems because it will shorten and simplify the language descriptions that are needed for parsing. It will also allow parser design to track new developments in linguistic theory.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's artificial intelligence research has been provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-80-C-0505. This is a revised version of a Ph.D. thesis proposal submitted to the Department of Electrical Engineering and Computer Science on February 28, 1984. Support for the author's graduate studies has been provided by the Fannie and John Hertz Foundation.

1. Preview

The focus of linguistic theory has shifted away from complex rule systems to modular systems of principles, but the practice of parser design has not kept pace. Natural-language parsers are still built on complex rule systems. Few implementation models are known for the new theories of grammar, and those that do exist fail to preserve their modular organization. Research is needed on how to embody the new theories in parsers.

1.1. Linguistic theory and parsing practice

The human ability to use and understand a language depends in part on knowledge of the syntactic structure of its sentences. Native speakers of the language learn its syntax by acquiring some mentally represented system of rules and principles. Their syntactic abilities result from possessing both such a grammar and implicit knowledge of how to put it to use. It is the business of generative linguistics to identify the rules and principles and explain how they are acquired and used.

A natural-language program is designed to approximate part of the human ability to use and understand natural language. Since it too must be sensitive to the syntactic structure of sentences, the program must be based on some approximation to the system of rules and principles that linguistics is striving to identify. Given this intimate connection between linguistics and the design of natural-language programs, it is natural to expect that parsing practice should closely track developments in linguistic theory. As linguistics provides better accounts of the rules and principles that define natural-language syntax, they can be embodied in programs that use better approximations to linguistic reality.

However, recent theoretical shifts in linguistics have not been matched by corresponding developments in the practice of parser design. Under early theories of transformational grammar, each language was described by a large system of complicated rules that meticulously spelled out the details of their operation. In contrast, new theories suggest that complicated language-specific rule systems do not form an important part of a person's syntactic knowledge. The focus of linguistic theory has shifted to the study of modular subsystems of grammatical principles and parameters.

1.2. Replicating the shift to modular syntax

A closer look at linguistic theory shows that there were good reasons for this shift. Early grammatical theories suffered from several scientific ills. Their detailed rule systems seemed derivative rather than fundamental. They were more stipulative than explanatory, they made weak claims about the nature of natural language, and they made language acquisition seem a mind-boggling task. The new modular theories cured these ills. By untangling the effects of separate underlying subsystems of grammar, they were able to come closer to uncovering the principles that form the true basis of syntactic knowledge.

A corresponding look at the practice of parser design confirms that no such shift has taken place there. Each language is still described by a large system of complicated rules that spell out the details of their operation. Such complicated rule systems are no more

desirable in engineering than they were in science, for the scientific ills that afflicted them in linguistic theory can translate to engineering maladies in parser design. They make it difficult to build natural-language systems.

The cure for these engineering maladies should be the same in parser design as it was in linguistic theory. It should be possible for a parser to base its actions on interacting principles and parameters instead of complicated rule systems. The development of such a parser would replicate in parsing practice the shift to modular theories of syntax. Just as the shift toward modularity simplified linguistic theories, it would shorten and simplify the descriptions of particular languages that are needed for parsers. It would thus make such descriptions easier to write.

1.3. A roadmap

Section 2 will sketch the logical relationship that binds together natural-language programs, theories of grammar, and linguistics. Section 3 will characterize the language descriptions that were used in old-style syntactic theories and point out their scientific disadvantages. Section 4 will show that the language descriptions used in current natural-language programs have largely the same character and a possess a corresponding set of engineering disadvantages. Section 5 will describe the theoretical shift in linguistics that cured the scientific ills of earlier theories, while section 6 will detail the proposal that the shift should be replicated in the design of natural-language systems. Section 7 will tentatively describe some possible design characteristics of a principle-based parser, and Section 8 will discuss the implementation technique of representing theoretical predicates and constraints implicitly in parser operation rather than explicitly in data structures. Section 9 will mention related earlier work, while section 10 will suggest a rough plan by which a principle-based parser might be developed.

2. The logical nature of natural-language parsing

The arrangement of words in a sentence matters as much as what the words are:[1]

(1) (i) Fred killed the spider

(ii) the spider killed Fred

(2) (i) fatal accidents deter careful drivers

(ii) deter drivers accidents fatal careful

(3) (i) I told Fred a ghost story

(ii) I told Fred a ghost story was the last thing I wanted to hear

A program that interprets sentences must know the syntactic structure of a language in

[1] The first four of these examples are from Baddeley (1976:310).

addition to the import of its words.[2,3] As one of its constituents such a program must have a *parser* that recovers the structure of input sentences.

In many cases the parser is a separate component that builds an explicit tree-like representation of syntactic structure. In other cases no explicit syntactic representation is built; the recovery of structure is intertwined with the process of semantic interpretation and the parsing component is only implicit. Either way, though, the parsing process analyzes an input sentence according to some theory of linguistic structure. The parser implicitly embodies this theory, supplementing grammatical knowledge with a way of using that knowledge to analyze sentences.

2.1. The definitive account of natural-language syntax

Any parser is implicitly based on a linguistic theory. Since only a theory of syntax can specify what syntactic structure the parser should assign to a sentence, it is a syntactic theory that defines the computational problem a parser must solve. The defining role of a syntactic theory makes the choice of syntactic theory important for parsing and natural-language processing.

Humans, not machines, speak definitive "natural language." The syntactic theory that is ultimately correct will be the one that succeeds at describing the tacit knowledge of linguistic structure that underlies a *human* speaker's syntactic abilities. Characterizing this tacit knowledge has long been a goal of generative linguistics.

According to linguistic theory, such knowledge takes the form of a mentally represented system of rules and principles that generate and relate various kinds of mental representations. Making up a mentally represented *grammar*, these rules and principles enter into various unconscious mental computations that are carried out in the process of producing and understanding sentences.

A natural-language program does not have to be based on the same system of rules

[2]On one possible account of the linguistic deficit involved in Broca's aphasia, the occasional comprehension difficulties of Broca's aphasics illustrate the importance of syntax. Lightfoot (1982:188f) comments:

> [Experiments] ... found that these patients could understand a sentence like *The apple that the boy is eating is red*, where the relations among the major words are constrained by our knowledge of the world: apples but not boys are red; boys eat apples, and not vice versa. A sentence like this can be understood without reliance on the function words and without having to analyze the structure of the sentence in any detailed way .... On the other hand, a sentence like *The girl that the boy is chasing is tall* is more difficult. Both girls and boys may be tall, and not only can girls chase boys, but also boys can chase girls. In order to understand such a sentence, one needs to be able to conduct a detailed analysis, identifying the proper role of function words like *the, that,* and *is*. This is beyond the capacity of Broca's aphasics, and they do not understand such sentences in the way that normals do .... In short, Broca's aphasics cope well with sentences where their knowledge of the world can get them by. They do very badly when they must rely on a syntactic analysis of the sentence in order to know what it means.

[3]Roger Schank, quoted in Winston and Prendergast (1984:166f), expresses quite a contrary view of the importance of syntax: "I think ... that research on syntax should have stopped fifteen years ago .... [S]yntax is not worth working on ...." If the natural-language systems that Schank advocates truly ignore syntax, they can be expected to have the kind of comprehension deficit that Lightfoot says Broca's aphasics have (see previous note).

and principles that is represented in the mind of a human speaker; only an approximation to that system is required. Indeed, the exact details of the human system are not currently known. Nevertheless, it is the mentally represented grammar that is the ultimate standard defining the language a person speaks. When the program grammar disagrees with the human grammar about the relation between sound (or orthography) and meaning, it is the human grammar that is correct.

If there is too much disparity between natural language and the version of natural language that a program accepts, the program's linguistic behavior can be so frustrating as to make it useless. A program with faulty "knowledge of language" can impose inappropriate interpretations on seemingly clear inputs. If its knowledge describes too few constructions, it can also place irritating, seemingly arbitrary restrictions on the range of syntactic forms it will accept.

2.2. The logical problem of parsing

Given a linguistic theory, how does it constrain the operation of a corresponding parser? The parser cannot simply read off the syntactic structure of a sentence from its surface form, since surface form does not explicitly indicate that structure. Rather, the parser must use its implicit knowledge of language in an active way to guide the recovery of syntactic structure.

From an abstract point of view, the task of the parser is to find a full syntactic representation that satisfies two conditions: the representation must be well-formed according to the linguistic theory that the parser embodies, and the surface form of the representation must be consistent with the input sentence. In many cases, it is likely that *two* possible syntactic representations will be well-formed under the theory and consistent with the sentence:

(4) visiting relatives can be boring

Therefore, two parses will be possible. Sentences like (4) will hence be syntactically ambiguous.

From the most neutral logical point of view, a theory of syntax does not constrain parser operation beyond this simple input/output relationship. Clearly, then, a theory of syntax does not completely determine a parser. In addition to knowing the possible syntactic structures of a language, the parser must possess an effective method of putting syntactic knowledge to use in actual sentence processing. An $LR(k)$ parser and an implementation of Earley's algorithm may both use the same context-free grammar and hence share the same linguistic knowledge. Although they will solve the same parsing problem, they will operate differently because they have different methods of putting their grammars to use in sentence processing. A theory of grammar is a theory of grammatical *competence*, while the operation of a parser also includes aspects of grammatical *performance*.

Marr (1982:25) distinguishes three levels at which an information-processing system must be understood. At the level of *computational theory*, it is necessary to identify the goal of the computation, understand why it is appropriate to the task at hand, and investigate the logic of the strategy by which it can be carried out. At the level of *representation and algorithm*, the relevant question is how the computation is implemented through the use of particular representations and algorithms. At the level of *hardware implementation*,

one investigates the physical realization of the representations and algorithms. In Marr's terms, the theory of syntax is part of the level of computational theory, while a complete description of the parsing process would include all three.

3. Language descriptions in early grammatical theories

Under early theories of transformational grammar, each language was described by a large system of complicated rules. The rules meticulously spelled out the details of their operation. Although these rule systems often described the facts about various constructions rather successfully, they failed to meet other scientific goals of linguistics:

- The reduction of grammatical phenomena to a complex, stipulative rule system did not have the explanatory power that reduction to a small set of principles could have.
- The choice of an unconstrained rule framework made excessively weak claims about the properties of human languages in general, since the availability of powerful descriptive devices in rules led to the ability to describe "languages" with properties quite unlike those attested in natural languages.
- The amount and complexity of the information required to describe individual languages made it a mystery how children could learn languages from the evidence available to them.
- The lack of substantial results from universal grammar made it a mystery what constraints a child might implicitly use to choose from the myriad possible grammars compatible with observed sentences.

As section 4.2 will show, the scientific disadvantages of such rule systems are not merely of theoretical interest. They carry over directly into problems for the designer of natural-language systems.

3.1. Complicated rule systems in early grammatical theories

Until recently, the rule systems involved in transformational theories of grammar were complicated and highly language-specific. Each rule explicitly spelled out the details of its application. For example, the Passive Transformation of English might have been stated as follows (Fiengo, 1977:36):

(5)

	X	NP	Y		V	NP	Z	
	1	2	3		4	5	6	
optional ⟹	1	5	3	be+en	4	*e*	6	by 2

The rule could apply to an underlying structure that roughly corresponds to the following surface sentence:

(6) the hippogriff loves the mermaid deeply

Operating on that structure as shown in Figure 1, the rule would produce the following sentence with accompanying structural information:

	X	NP	Y			V	NP	Z	
	1	2	3			4	5	6	
optional ⟹	1	5	3	be+en		4	*e*	6	by 2

(a)

[$_S$ [$_{NP}$ The hippogriff] *pres* [$_V$ love] [$_{NP}$ the mermaid] deeply]

(b)

the hippogriff	*pres*	love	the mermaid	deeply	
X	Y	V	NP	Z	
1	2	3	4	5	6

(c)

1	5	3	be+en	4	*e*	6	by	2	
	the mermaid	*pres*	be+en	love	*e*	deeply	by	the hippogriff	

(d)

Figure 1: The old-style Passive transformation (5), here repeated as (a), would transform the structure associated with string (6) into the structure associated with string (7). The underlying structure (b) would match the rule condition as indicated in (c). The correspondence established by matching would then be used to build an output structure as shown in (d). Various theoretical details and the treatment of Tense have been glossed over in this example.

(7) the mermaid *pres* be+en loved *e* deeply by the hippogriff

(The symbol *e* refers to the empty constituent.) Other minor rules would apply to give the passive marker *-en* and the tense *pres* their proper expression, and the passive version of the sentence would emerge:

(8) the mermaid is loved deeply by the hippogriff

The statement of the Passive rule (5) is quite complicated. The condition of the rule uses both variables such as X and categories such as V to describe the surrounding context; the action of the rule includes two movements, the insertion of an empty category, and the insertion of *be*, *-en*, and *by*. This complexity is still not enough, however; since this rule creates a *by*-phrase, some other rule will be needed for producing passives that do not contain *by*-phrases:

(9) the temple was destroyed in 1945

The rules defining basic constituent structure were also complicated and idiosyncratic; they explicitly specified such details as constituent order and type. For example, Jackendoff (1977) proposed the following phrase-structure rules to describe basic constituent order

within a sentence:

$$
\begin{aligned}
& V''' \Longrightarrow (N''')\ (M''')\ V'' \\
(10) \qquad & V'' \Longrightarrow (have - en)\ (be - ing)\ ([Adv, +Trans]''')^*\ V'\ (P''')^*\ (\overline{S}) \\
& V' \Longrightarrow V\ (N''')\ (Prt''')\ ([-Obj, -Det]''')\ (P''')\ ([+Obj, +Comp]''')
\end{aligned}
$$

Here parentheses around a constituent indicate optionality, the asterisk indicates indefinite repetition, and square brackets indicate feature notation. Jackendoff's V' rule could apply to generate the verb phrase in this sentence:

(11) the judge $[_{V'}$ $[_{V}$ sent] $[_{N'''}$ the convict] $[_{P'''}$ to prison]]

Jackendoff also used other notational devices such as angle brackets and curly braces in the statement of phrase-structure rules.

3.2. Scientific disadvantages of complicated rule systems

The rule systems found in early grammatical theories had to be complicated to operate properly. A powerful descriptive apparatus was necessary for writing down the restrictions. Both of these facts led to unfortunate consequences.

3.2.1. Detailed rules seemed descriptively necessary

Early generative grammarians were trying to carefully and precisely formulate rules that could describe the properties of various grammatical constructions. In pursuing this goal they were driven to write very detailed rules, for there seemed no other way to prevent the rules from applying improperly. For example, the Passive rule (5) had to introduce the copula *be* and the passive morpheme *-en* so that passive constituent order couldn't surface with active verb forms:

(12) *the mermaid loves deeply by the hippogriff

It had to mention V so that the proper insertion position for *be+en* could be specified. Adjacency to V was also required so that other ungrammatical sentences wouldn't be generated:

(13) (i) John hit Bill with a club

(ii) *a club was hit Bill with

Even with the detailed rule, some unwanted cases might slip through depending on how the other rules worked:

(14) (i) projects like that, he'll never get ME to support

(ii) *me, he'll never get e to support by projects like that

And in any case, another rule would be needed for generating agentless passives such as (9). The theory would thus fail to capture any similarity between "long" and "short" passives.

3.2.2. Complex rule systems are not explanatory

The early grammatical theories were fairly successful at using systems of rules to capture the properties of various constructions, but the rule systems were highly stipulative.

The theories stated the rules, but could give no theoretical reasons *why* the details of the rules should be the way they were. For example, the statement that a rule is *obligatory* is merely descriptive, leaving unexplained the question of why a derivation in which it fails to apply results in ungrammaticality.

Science is generally not content to leave complexity unexplained, saying the complexity is "just the way things are," but always strives to explain it through reduction to simpler principles. There was the possibility that the complicated rule systems were only *derivative*, corresponding to the combined effects of more fundamental principles rather than being fundamental in themselves. If that turned out to be the true situation, the early theories could still be partly correct. The general processes that they took to be involved in the derivation of various constructions could still be involved, but with the details of their operation following from general principles rather than the details of rule statements.

3.2.3. Complex rule systems are too unconstrained

The rule systems also drew on a powerful, unconstrained descriptive apparatus. In attempting to restrict the application of rules to their proper domains, grammarians used a wide variety of notational devices in the rule patterns or *structural descriptions* (SDs) of rules:

> Among the enrichments of the theory of SDs that appear in the literature, theoretical and applied, are the following: disjunctions of [SDs], meaning that the factors may satisfy any one of the disjuncts; wider possibilities for [individual elements of rule patterns]; SDs defined in terms of Boolean conditions [on the set of SDs applying to the sentence]; conditions expressed in terms of quantifiers; conditions involving grammatical relations [such as subject and object]; SDs expressing quite arbitrary conditions on phrase markers or even sets of noncontiguous phrase markers of a derivation; SDs expressing conditions not limited to a single derivation; SDs involving extrasyntactic or even extragrammatical factors, e.g., beliefs. (Chomsky, 1976:310)

If linguistic theory makes available without constraint such a wide variety of mechanisms for use in language descriptions, it will make extremely weak claims about what constitutes a possible natural language. Unless the descriptive apparatus is further constrained, the theory of universal grammar will be scientifically vacuous because it will claim virtually nothing.

A weak theory of universal grammar is thus undesirable on general scientific grounds. However, it is further undesirable because it is incorrect: it makes the wrong predictions about the range of variation in natural languages. A weak theory predicts that natural languages can potentially differ arbitrarily much in structure, but this wide range of variation is not attested.

For example, as Baltin (1981:4) notes, Wackernagel's Law states that a phenomenon called *cliticization* always places clitics either in second position in the sentence or attached to the verb. There are apparently no languages in which clitics attach to the last noun

phrase in the sentence, or to the third word ignoring constituent boundaries. Greenberg (1963) cites other simple examples of regularities among languages.

3.2.4. Complex rule systems make a mystery of language learning

A fundamental problem of generative linguistics is to discover the form and content of the knowledge that a person acquires when learning a language. The hypothesis that this knowledge takes the form of a system of complicated rules, complete with information about the order in which they must apply and about whether they are obligatory or optional, makes it hard to understand how a children could ever learn their native tongues. Stowell (1981:64) notes this problem in connection with the linguistic theories of the sixties:

> The very complexity and variety of the transformational grammars of individual languages frustrated attempts to develop explanatory theories of language acquisition. Although there were some promising possibilities of formal linguistic universals, most of the complexities in specific grammatical rules appeared to be tremendously idiosyncratic. This was perhaps most obvious for the transformational rules, each of which appeared to require an arbitrary collection of elementary operations ... and various mysterious conditions preventing individual rules from applying in certain environments. It was obvious, from the perspective of a reasonable theory of acquisition, that these complexities could not be directly learned on the basis of experience, since the learning task would have to depend on explicit negative evidence of a very obscure kind .... On the other hand, very few of the observed conditions could be deduced from known properties of the language faculty, leading Chomsky [(1965:46)] to remark that "no present-day theory of language can hope to attain explanatory adequacy beyond very restrictive domains."

With detailed systems of language-particular rules, there are just too many details in the description of a language for the language learner ever to acquire it.

3.2.5. An unconstrained framework makes learning impossible

Language acquisition requires the learner to construct a grammar on the basis of finite evidence. The grammar can apply to an indefinitely large range of sentences not heard before. If the language learner is to be successful, the constructed grammar must agree with the grammars of others in the speech community.

The language learner cannot succeed if armed only with very weak constraints on what the structure of the target language might be like. There are just too many ways to project beyond experience. In a sufficiently powerful descriptive framework, for instance, there are indefinitely many grammars compatible with any finite amount of linguistic experience. The language learner must use some principle of universal grammar to choose among them. Without such a principle, the learner may not choose a grammar that agrees with those of others.

An unconstrained framework with a weak theory of universal grammar gives the lan-

guage learner almost no guidance about how to solve the problem of projecting beyond a finite range of observed evidence. Language learning under such circumstances is impossible. More restrictive theories are necessary in order to explain language acquisition.

4. Language descriptions in natural-language systems

Modern syntactic theories have found a cure for the scientific ills of section 3.2, and section 5 will describe it. First, however, this section will establish that the language descriptions that underlie existing natural-language parsers have many of the same problems that beset early syntactic theories. Parser design could benefit from the same curative measures that improved linguistic theory.

The grammars that are embodied in most existing parsers consist of complex, language-dependent rule systems that explicitly spell out such matters as the orders and types of constituents in various constructions. The practice of natural-language parsing is thus in roughly the same situation as early linguistic theory: each language is described by a large set of complicated rules that exhaustively specify the details of their application. In much the same way that complicated, language-dependent rule systems fall short of the scientific goals of linguistics, they make it difficult to meet the engineering goal of constructing natural-language systems:

- Describing grammatical phenomena by means of a complex, stipulative rule system instead of reducing them to underlying principles leaves unanswered the question of why the details of the rule system are the way they are. Without principles that explain why the details should be one way rather than another, the system designer is just as likely to get them wrong as right.

- The choice of an unconstrained rule framework makes weak claims about what natural languages are like. The unrestricted availability of powerful descriptive devices gives the system designer the unwanted ability to describe "languages" with properties quite unlike those attested in natural languages.

- When the descriptions of individual languages are large and complex it is something of a mystery how a system designer can ever succeed at building a parser. Surely this notoriously difficult task could be easier with a more concise characterization of the differences among languages.

- Large grammars can also make natural-language systems run slowly.

- Like the language learner, the system designer must arrive at a rule system that projects beyond the example sentences that shaped its design. The failure to seek guidance from universal grammar leaves the designer without constraints to aid in choosing from the myriad possible language descriptions that will work properly on simple examples.

4.1. Complicated rule systems in existing natural-language programs

Whether it is an augmented transition network, an augmented context-free grammar, or a set of pattern-action rules, the rule system that encodes the linguistic knowledge of a current natural-language system is likely to be large, complicated, and highly language-dependent. A few examples will illustrate.

4.1.1. Existing parsing rules are complicated

The language descriptions that underlie existing natural-language parsers are made up of complex rules that generally spell out the details of their application quite specifically. Like Jackendoff's phrase-structure rules (10), even unadorned context-free rules spell out the order, type, and obligatoriness of constituents in various constructions. Most systems, however, spell out much more.

Robinson (1982:42), for instance, cites the verb-phrase rule shown in Figure 2 as typical of the rules used in a system for interpreting English dialogue. (Not surprisingly given the complexity of this rule, transcription errors appear to have affected parenthesis matching in the published version.) ATN-based systems also use detailed tests and actions on grammar arcs; see Figure 3.

Even Marcus (1980), who constrains the information available to parsing rules, uses some rather complicated tests and rule-packet activations that tell the parser what constituents to expect and where to attach them. Figure 4 illustrates. Marcus's framework also requires the parser designer to notice potential ambiguities in the interpretation of surface cues, writing diagnostic rules to decide between competing possible parser actions. A diagnostic rule for a construction might be considered the most detailed rule of all, since it requires the parser designer to consider not only the construction at hand, but all other constructions that might look similar given the limited information available to the parser at various points. Marcus's diagnostic rules also tend to require access to a wider range of information than other grammar rules. Figure 5 gives examples.

4.1.2. Existing rule systems are large

In addition to being detailed, the description of a language that underlies a typical parsing system is lengthy. A typical ATN system has several hundred arcs; for instance, Bates (1978:238) mentions one with 83 states, 202 arcs, and 386 actions. Robinson (1982:27) explicitly describes the DIAGRAM augmented phrase-structure grammar as "large and complex," and the set of verb-phrase rules in that system (:45f) seems to bear out that description. The rules are shown here in simplified form:

(15)
```
VP = V (NP1 ([NP2 / P]))
VP = V P (NP)
VP = V (NP) ("THAT") SDEC
VP = V (NP) INFINITIVE
VP = V (NP) [PPL VP / ADJP]
VP = V (NP) (ING) [VP / BE PRED]
```

```
(VP1 VP = V (NP1 (NP2 / P));
     CONSTRUCTOR (PROG ((PARTICLE (@ DIAMOND.SPELLING P)))
                       (COND
                         [(@ NP1)
                              (OR (@ DIROBJ V)
                              (F.REJECT (QUOTE F.DIROBJ)))
                       (COND
                         ((@ NP2)
                          (OR (@ INDIROBJ V)
                              (F.REJECT (QUOTE F.INDIROBJ))))
                         ((@ P)
                          (OR (FMEMB PARTICLE (@ PARTICLE V))
                              (F.REJECT (QUOTE F.PARTICLE)))
                          (AND (@ PRO NP1)
                               (@FACTOR (QUOTE F.PARTICLE)
                                        LIKELY))
                          (COND
                            ((@ NCOMP NP1)
                             (OR (@ NP NCOMP NP1)
                                 (@FACTOR (QUOTE F.PARTICLE)
                                          UNLIKELY)
                                 (AND (@ NCOMP NP NCOMP NP1)
                                      (@FACTOR (QUOTE F.PARTICLE)
                                               UNLIKELY))))
                            (T (@SET BAREV T)
                               (@FROM V DIRECTION DIROBJ))))
     TRANSLATOR (PROGN [COND
                         ((@ NP)
                          (@SET ROLE (QUOTE DIROBJ) NP2)
                          (@SET ROLE (QUOTE INDIROBJ) NP1)
                          (@SET SEMANTICS (COMBINE
                                (@ SEMANTICS V)
                                (@ SEMANTICS NP2)
                                (@ SEMANTICS NP1)))
                        (T (AND (@ NP1)
                                (OR (@ INDIROBJ V)
                                    (@SET ROLE (QUOTE DIROBJ) NP1))
                                (@SET SEMANTICS (COMBINE
                                                  (@ SEMANTICS V)
                                                  (@ SEMANTICS NP1))))
```

Figure 2: This verb-phrase rule from the DIAGRAM system of Robinson (1982) is complex and detailed.

```
(VP/V
  (CAT V (AND (GETF PASTPART)
              (EQUAL (GETR V) (QUOTE BE)))
       (HOLD (QUOTE NP) (GETR SUBJ))
       (SETRQ SUBJ (NP (PRO SOMEONE)))
       (SETR AGFLAG T)
       (SETR V *)
       (TO VP/V))
  (CAT V (AND (GETF PASTPART)
              (EQUAL (GETR V) (QUOTE HAVE)))
       (ADDR TNS (QUOTE PERFECT))
       (SETR V *)
       (TO VP/V))
  (CAT V (AND (GETF UNTENSED)
              (GETR MODAL)
              (NULLR V))
       (SETR V *)
       (TO VP/V))
  (CAT V (AND (GETF PRESPART)
              (EQUAL (GETR V) (QUOTE BE))
       (ADDR TNS (QUOTE PROGRESSIVE))
       (SETR V *)
       (TO VP/V))
  (JUMP VP/HEAD T
       (COND ((OR (GETR MODAL) (GETR NEG))
              (SETR AUX (BUILDQ ((@ (AUX) + +))
                                MODAL NEG))))))
```

Figure 3: This simplified ATN state forms part of the verb-phrase network in a grammar described by Bates (1978:208). Like the rule in Figure 2, it is complex and detailed.

```
VP = V (NP) [WHPP / WHNP / WHADJP] [SDEC / INFINITIVE]
VP = VP (",") [PP / INFINITIVE / ADVP]
```

Marcus's (1980) parser is somewhat smaller; one version has 101 rules, and many of these rules pertain to numbers, dates, and other idiosyncratic elements of his parsing application. In part, the smaller size of Marcus's parser derives from the fact that it is more closely related to transformational accounts of grammar than to accounts that use phrase-structure rules to describe surface configurations directly. (See Marcus, Chapter 5.)

4.1.3. Existing rule systems are language-dependent

The highly language-dependent character of the above-cited systems should be clear from the sample rules given. Surely the details of what to expect at various points in a parse would change when going from English to a verb-final language, a postpositional language, or a language with no ambiguity between prepositional phrases and infinitives.

Naturally, any rule system that expresses knowledge of a particular language must change from language to language. The unfortunate characteristic of existing rule systems is not that they differ from language to language, but that they differ more than the language structures do. Existing parsers do not seem to be modularized in such a way that changing a single language characteristic corresponds to changing a single part of the language description. Since small changes in underlying parameters can have large effects

```
{RULE main-verb PRIORITY: 15. IN PARSE-VP
[=verb] -->
Deactivate parse-vp.
If c is major then activate ss-final else
if c is sec then active emb-s-final.
Attach a new vp node to c as vp.
Attach 1st to c as verb.
Activate cpool.
If there is a verb of c and it is passive
    then activate passive; run passive next.
If it is inf-obj then
    if it is to-less-inf-obj then activate to-less-inf-comp andthen
    if it is to-be-less-inf-obj then activate to-be-less-inf-comp andthen
    if it is 2-obj-inf-obj then activate 2-obj-inf-comp
       else activate inf-comp;
          if it is subj-less-inf-obj then activate subj-less-inf-comp
          else if it is no-subj then activate no-subj.
If it is that-obj then activate that-comp.
If there is a WH-comp and it is not utilized
    then activate WH-vp else
if the current S is major then activate ss-vp else
activate embedded-s-vp.}

{RULE WH-WITH-NP-PP-NEXT PRIORITY: 7 IN WH-VP
[=np] [=prep] -->
If the greatest possible number of objects of c is greater than 1
    and a prepositional phrase of 2nd and the WH-comp
        fits a pp slot of c
    or
    the greatest possible number of objects of c is equal to 1
    and a prepositional phrase of 2nd and the WH-comp
        fits a pp slot of the current s
    then run objects next else
If the greatest possible number of objects of c is greater than 1
    then run wh-with-np-next next else
Run too-many-nps next.}

{RULE WH-WITH-PP-NEXT PRIORITY: 5 IN WH-VP
[=prep] [=np] -->
If a prepositional phrase of 1st and 2nd fits a pp slot of c
    then run pp next else
If it isn't true that
    a prepositional phrase of 1st and the WH-comp
        fits a pp slot of c
    then if the greatest possible number of objects of c
        is greater than 0 then run create-wh-trace next
        else run too-many-nps next
    else
If the lowest possible number of objects of c is greater than 0
    then run create-wh-trace next else
    run wh-pp-build next.}
```

Figure 4: These rules from Marcus (1980) illustrate that the rule-packet structure of the parser can be somewhat intricate and the rule actions can be complicated.

```
{RULE HAVE-DIAG PRIORITY: 5 IN SS-START
[=*have, tnsless] [=np] [t] -->
If 2nd is ns, n3p or 3rd is not verb, or 3rd is tnsless
then run imperative next else
run yes-no-q next.}

{RULE WHICH-DIAGN IN CPOOL
[=*which; * is not any of  quant, relpron] -->
If the np above c is np, modible then
label 1st pronoun, relpron, wh
else label 1st quant, ngstart, ns, npl, wh.}

{RULE THAT-DIAG-1 IN CPOOL
[=*that; * is none of comp, det, pronoun] [=np] -->
If there is not a det of 2nd
    and there is not a qp of 2nd
        and the nbar of 2nd is none of npl, massn
and 2nd is not not-modifiable
then attach 1st to 2nd as det; label 1st det, ns
else if c is a nbar then label 1st pronoun, relpron
else label 1st comp.}

{RULE THAT-DIAG-3 PRIORITY: 5 IN CPOOL
[=*that; * is none of pronoun, comp] [=np]
[**c; the verb of the vp of the current s is that-obj;
 the lowest possible number of objects of the current s
 is equal to 2] -->
Label 1st comp.}
```

Figure 5: In the framework of Marcus (1980), diagnostic rules such as these decide between different possible parsing actions when the normal grammar rules are not sufficient to determine what to do next.

on the surface distribution of constituents, it is not surprising that parsing rules should be highly language-dependent when they spell out the details of their surface application.

Subject-verb agreement provides one example. Suppose an ATN parser checks agreement by storing grammatical features of the subject in a register and later comparing them to features of the verb. If the parser is to be adapted to parse a verb-initial language, in addition to rearranging arcs it will be necessary to swap the register store and register comparison operations. To take another example, the mechanism that Marcus (1980) uses to construct noun phrases relies heavily on the fact that English noun phrases are determiner-initial and hence determiners will be encountered first in a left-to-right scan. Adapting the Marcus parser to a determiner-final language could require substantial revision.

4.2. Engineering disadvantages of complicated rule systems

Many of the scientific disadvantages that afflicted complex, language-specific rule systems in linguistics translate into engineering disadvantages that afflict similar rule systems in the realm of natural-language processing. They help make parser design a difficult task.

4.2.1. Detailed rules might seem descriptively necessary

Detailed rules may seem necessary to the designer of a natural-language system just as they seemed necessary to early grammarians. After all, *something* must account for surface complexity. In a parser built on context-free rules, for example, it is necessary to have a complicated rule system. Context-free rules directly spell out the surface orders and types of constituents in various constructions, so they must reflect surface complexity in rather direct fashion.

There is an alternative, however. Modern linguistic theory accounts for surface complexity by invoking the combined operation of several independent systems of principles. If parsing were based around such principles rather than explicit rules, there might be no need for detailed rules in describing the "core grammar" of a language.

4.2.2. Complex rule systems are not explanatory

A linguistic theory that describes grammatical phenomena by means of a complex, stipulative rule system instead of reducing them to underlying principles is at a scientific disadvantage because it does not explain *why* the details are the way they are. This disadvantage applies in the engineering domain as well. Without principles to explain why the details of rules should be one way rather than another, the designer can easily get them wrong.

4.2.3. Complex rule systems are too unconstrained

It is a theoretical advantage for a theory to place strong limits on the allowable set of rules of grammar, since a theory that places weak limits says very little about the nature of language. Once again, this theoretical advantage translates into a practical one. It would be easy to construct a language description for use in a parser if the grammatical framework provided so many constraints that the parser designer was left with no choice but to write the correct grammar! Correspondingly, it is very difficult to write a grammar when the grammatical framework is completely unconstrained, giving no clues at all about the properties of the correct grammar.

A somewhat frivolous example may help to illustrate the point. In an unconstrained parsing system, the grammar writer is given complete freedom to write the grammar according to personal choices. There is nothing to stop the parser designer from writing rules that are sensitive, say, to whether the number of words processed so far in the input sentence is prime.

Such freedom is an advantage to a programmer who intends to write a prime-number generator, but it is a hindrance to the designer of a natural-language system. Rules about prime numbers are not needed for parsing any natural language, so the major freedom granted is the freedom to make mistakes.[4]

[4] In a sense, the search for a restrictive parser-writing framework is thus similar in spirit to the effort within computer science to design computer languages that do not allow certain kinds of erroneous programs to be expressed. In both cases, there is an attempt to fit the framework rather exactly to the range of problems to be solved. Just as prime-number rules are not needed for describing natural-language syntax, programs that apply operations to inappropriate data types are not needed for useful programming applications.

In reality there is little danger that the parser designer will accidentally write into the grammar a dependence on prime numbers, but there is a danger that the designer will write in conditions that are unnatural in other ways. The more constraints a theory can offer, the more guidance it offers the grammar writer; the more constraints the better, so long as the constraints do not rule out the correct grammar for the language at hand. A tightly constraining theory of grammar makes the grammar writer's task easier.

A restrictive theory of universal grammar can also expand the available range of parser implementation options. The more specific the restrictions on grammars, the greater the probability that special properties of grammars may allow them to be efficiently processed or perspicuously implemented. To take an example outside the domain of natural language, finite-state automata can be simulated more simply if they are known to be deterministic than if they may be nondeterministic.

4.2.4. Complex rule systems make describing particular languages difficult

A linguistic theory that hypothesizes large systems of language-specific rules as the basis for the native speaker's knowledge of language is at a scientific disadvantage because it cannot account for the ease with which children acquire their languages. The description of a language takes too many details.

This disadvantage also operates in the engineering domain. The difficulty of writing a language description for a parser can be expected to grow as the description gets larger. The parser designer cannot easily understand an ATN system with hundreds of states and thousands of arcs. Just as concise characterizations of the syntactic parameters along which languages differ make it possible to approach the goal of explaining language acquisition, they can make it easier for the parser designer to specify the differences among French, Italian, and Warlpiri.

4.2.5. Complex rule systems can slow down parsers

The size of the underlying rule system figures in the running time of many parsing algorithms. Earley's (1970) algorithm for parsing context-free grammars, for example, can quadruple its running time when grammar size is doubled.[5]

4.2.6. An unconstrained framework makes system extension difficult

Explaining how a language can be acquired on the basis of finite linguistic experience is a major theoretical goal of linguistics. The language learner cannot succeed given only weak constraints on what the structure of the target language could be like. In an unconstrained framework, an indefinitely large number of grammars will be compatible with any finite amount of linguistic experience. Few of these grammars will yield appropriate results when applied to sentences not heard before.

[5] This argues against Fodor's (1983) claim that modular systems of grammar lead to less efficient parsers. Fodor claims that a parser based on a modular theory will be at a disadvantage because it must access and integrate information from more than one source. On the one hand the possibility of limited parallelism can vitiate that objection, while on the other hand the combinatorial effects involved in a non-modular system can increase its size enough to make it run more slowly rather than faster.

The designer of a natural-language system faces a problem that in a few respects is similar to that of the language learner. Any language description that the designer constructs will project in some way beyond the example sentences that shaped its design. In an unconstrained framework, the designer can choose from a multitude of systems that will work properly on the examples that have been considered at a given point. Only a few, however, will also apply properly to complex examples. An unconstrained framework gives the designer no help at making a felicitous choice. It treats more or less equally the different possible ways of projecting beyond the examples considered so far.

The ultimate possibility of explaining language acquisition shows how far a restrictive theory of universal grammar could in principle go toward making the task of language description easier. With language acquisition well-understood, a mechanical algorithm might be implemented that could acquire the syntax of a natural language through exposure to its sentences. The task of writing a syntactic description of the target language would then be trivialized.

5. The shift to modular theories of grammar

Sections 3.2 and 4.2 have shown that language descriptions made up of large systems of detailed rules have both scientific and engineering disadvantages. Modern linguistic theory has cured those scientific ills by shifting from the study of complex rule systems to the study of modular subsystems of grammatical principles and parameters.

Rules still exist, but the rule systems are increasingly regarded as simple and impoverished. No longer does each rule meticulously spell out the details of its application; rather, the conditions of proper rule application are determined by general principles that constrain linguistic representations. Many of the principles are universal and hence are not stated in the descriptions of particular languages.

The new modular theories of grammar solve many of the problems that were associated with earlier theories:

- They provide better explanations of many grammatical phenomena by reducing them to a small set of principles rather than a complicated, stipulative rule system.
- They allow universal grammar to place strong constraints on the possible range of "core syntactic rules" since they do not require the details of rule application to be stated in the rules themselves.
- They reduce the mystery of language acquisition by condensing the basic syntactic description of an individual language down to a set of values for a small list of parameters. Grammars are no longer huge and complicated.
- The strong constraints that they place on possible grammars simplify the language learner's problem of choosing from the possible grammars compatible with observed sentences. The number of possible grammars is no longer astronomical.

5.1. The scientific benefits of modular theories

The surface behavior of a system that is composed of interacting components usually presents a bewildering array of complexity. When such complexity shows up in the theory of a system as well, it is often a symptom that theory has not yet penetrated to the true underlying principles that govern system operation. A theory that needs epicycles upon epicycles may be describing derivative effects rather than fundamental laws.

Recent linguistic theories attempt to understand the apparently complex properties of various constructions as arising from interaction among different principles and grammatical subsystems. When such modular theories are possible, they can be expected to have scientific advantages. Through the process of untangling separate effects they are able to reduce to simpler principles many of the complicated stipulations that would otherwise seem necessary.

5.1.1. Modularity yields brevity and simplicity

According to a modular theory of grammar, surface linguistic phenomena result from the interaction of independent subsystems of grammatical rules and principles; the components of different subsystems typically have different functions and properties. According to a non-modular theory, surface linguistic phenomena result from the operation of a single, unitary rule system; grammatical rules are of the same type throughout.

When the phenomena at hand admit a modular description, a modular theory will be simpler than a non-modular theory. If it is possible to describe the phenomena in terms of separate subsystems acting independently, then a modular theory can simply describe the separate subsystems. A non-modular theory, however, must describe the combined surface effects because it refuses to untangle the separate underlying factors.

It is easy to find examples of how the choice of a non-modular theory over a modular one can cause grammar expansion. For example, the grammar gets larger when phenomena such as subject/verb agreement and apparent movement of displaced constituents are handled in the same rule system that defines the basic constituent structure of the language. Consider a non-modular grammar that handles subject/verb agreement by multiplying the number of rules and nonterminals in the grammar, using such rules as $S \Longrightarrow NP_{sg}\ VP_{sg}$ and $S \Longrightarrow NP_{pl}\ VP_{pl}$. Such a grammar will be larger than a modular grammar that treats subject/verb agreement by superimposing agreement rules on a simpler grammar that ignores agreement.

5.1.2. Modularity yields tighter constraint

Since a modular theory separates subsystems that have different properties, a modular theory can also make stronger claims than a non-modular one. Suppose a modular theory postulates subsystems of rules of types A and B, while a non-modular theory uses only a single rule type C. Rules of type A and type B must have somewhat different properties, or there is no reason to place them in different subsystems. It is unavoidable, then, that the statements that can be made about rules of type C must be weaker than the statements that can be made about rules of types A and B. Certain generalizations are necessarily lost

in going to rules of type C, since two mechanisms with different properties have been made to look alike; the properties that distinguish rules of type A from type B cannot be true of all rules of type C. As usual, increased generality yields weakened constraint.

Early transformational grammars, for instance, used the single mechanism of transformations to describe both the reference of pronouns and the displaced position of *wh*-words in questions. More recent theories assign the treatments of these phenomena to separate grammatical components. A transformation handles *wh*-movement, but interpretive rules handle pronominal reference. Chomsky (1976) notes that once transformations and interpretive rules are separated, they can be seen to have different properties and obey different sets of constraints. It is possible to tighten the range of possible transformations as well as the range of possible interpretive rules.

5.2. Factoring constraints out of grammatical rules

Factoring general constraints out of syntactic rules simplifies grammatical theory because constraints do not have to be repeatedly stated in the conditions of rules to which they apply. The simplification of individual rules also reduces the number of rules needed because many rules that were distinct in earlier theories turn out to be the same when cluttering details are removed.

5.2.1. Transformations have been reduced to simple forms

Generative grammarians have long sought to discover the restrictive set of conditions on possible rules of grammar that allows the language learner to converge on the correct grammar based on limited evidence. The shift from rules to principles has its historical roots in the quest to reduce the possible variety of transformations.

Chomsky (1976) proposed to impose on the structural descriptions of transformations a condition that would restrict the use of categorial symbols such as NP. An SD would not be allowed to mention two successive categorial symbols unless one or the other represented a constituent changed by the rule. In particular, the following detailed SD for Passive would be ruled out:

(16) X NP Aux V NP by Δ X

(Here Δ is a dummy marker.)

Under Chomsky's proposed restriction and some additional assumptions, the SDs for the main operations involved in the derivation of passive sentences would have a simpler form instead:

(17) X NP X NP X

This line of argument eventually led to a very general formulation of the movement rule involved:

(18) Move NP

The movement involved in Passive was thus seen as one manifestation of a rule that says "move any NP anywhere" rather than the result of a rule with a detailed context of application.

When traditional transformational rules were simplified in so drastic a fashion, many that had previously been considered distinct collapsed into one. For example, the traditional rule of Raising to Subject came out as just another manifestation of Move NP:

(19) (i) [$_{NP}$ *e*] seems the bear to be hungry

(ii) the bear seems *e* to be hungry

Another whole collection of grammatical processes came out as instances of another simple rule called Move-*wh*.

In the new theories, traditionally distinct grammatical processes were thus regarded as formally identical. They no longer corresponded to the operation of separate rules:

> The notions "passive," "relativization," can be reconstructed as processes of a more general nature, with a functional role in grammar, but they are not "rules of grammar." (Chomsky, 1981:7)

It was clear that the complexity of the transformational component would be reduced if transformations had the simple and general character illustrated in (18) rather than the detailed, one-rule-per-process character of the old rules such as (5).

5.2.2. Constraints rule out misapplication

As Chomsky realized, however, rules such as Move NP overgenerate massively unless restricted in some way. Consider this "derivation," for instance:

(20) (i) John saw Bill

(ii) Bill saw *e*

Why can't Move NP turn (20i) into (20ii)? If it were to turn out that some *ad hoc* condition would be required in order to prevent such derivations, there would be no advantage to "simplifying" rules down to minimal form. Complexity would simply be shifted from one part of the grammar to another.

No *ad hoc* conditions are required, however. Most of the "bad" movements are ruled out by conditions that have independent justification. For example, the above movement is ruled out independently by several different general conditions in modern linguistic theory. One of the simplest is the principle of *recoverability of deletion*,[6] which among other consequences forbids a rule from moving a constituent into a position that already has another constituent in it. In the above case, recoverability of deletion forbids moving *Bill* on top of *John*.

Other misapplications of Move NP are ruled out by other independently motivated principles of grammar. Modern theories factor out general constraints, maintain simple formulations of transformational rules, and thus achieve two simplifications. A grammar is simplified when a general condition is stated once rather than many times in many rules, and it is shrunk when the removal of detailed specifications from rules causes previously distinct rules to fall together.

[6] The recoverability principle itself was once stated in individual rules rather than factored out as a separate constraint. See Lasnik (1976:3).

5.2.3. GB-theory uses modular subsystems of principles

Modern transformational theory goes by the name of *government-binding theory,* or GB-theory, because the technical notions of government and binding play a central role. Current GB-theory[7] postulates four grammatically significant levels of description. The level of *D-structure* expresses the assignment of *θ-roles* such as Agent-of-Action to appropriate constituents. A D-structure position may not exist unless "licensed" in one of a few ways. D-structure configurations are also constrained by *X-bar theory,* which is concerned with the structural relationships between the "head" of a phrase and its various satellites.

D-structure is converted to *S-structure* through the operation of rules of the form Move α, where α is a constituent. (Move NP is one subcase.) Movement leaves behind an empty *trace* associated with the moved constituent. S-structure is essentially an enriched version of ordinary surface structure. S-structure representations are mapped independently to representations in the *LF* (logical form[8]) and *PF* (phonetic form) components. As currently conceived, the level of LF functions largely to indicate the scope of quantifiers and similar elements. Various conditions restrict the relationship between a quantifier and its bound variables at LF. The *Empty Category Principle* also places requirements on the distribution of empty categories at LF.

The *θ-criterion* applies at all linguistic levels and requires (roughly) that each noun phrase be associated with one and only one *θ*-role. Since the chain formed by a moved constituent and its traces is assigned *θ*-role as a unit, the *θ*-criterion acts as one constraint on movement. The *Projection Principle* requires representations at various levels to be fundamentally just projections of lexical items, in the sense that the properties of lexical items (such as whether or not a verb is transitive) must be represented at each linguistic level.

Chomsky (1981) briefly describes several subsystems of principles in an introductory passage:

> The subsystems of principles include [bounding theory, government theory, *θ*-theory, binding theory, Case theory, and control theory]. Bounding theory poses locality conditions on certain processes and related items. The central notion of government theory is the relation between the head of a construction and categories dependent on it. *θ*-theory is concerned with the assignment of thematic roles such as agent-of-action, etc. (henceforth: *θ*-roles). Binding theory is concerned with relations of anaphors [referentially dependent elements such as "each other" and NP-trace], pronouns, names, and variables to possible antecedents. Case theory deals with assignment of abstract Case and its morphological realization. Control

[7] This greatly condensed summary is based on Chomsky (1981) and on Chomsky's Fall 1983 class lectures.

[8] "Logical form" is used as a technical term within GB-theory. In this context, the ordinary meaning of the term is only suggestive and can be misleading. Representations at the LF level do not carry all of the information that is relevant to logical form in other senses. For example, the LF representation of a sentence according to GB-theory is not directly relevant to determining the logical validity of inferences that might be drawn from the sentence; similarly, the occurrence of a quantified variable at the LF level carries no ontological commitment, despite the famous dictum that to be is to be the value of a bound variable. See Chomsky (1981:17).

> theory determines the potential for reference of the abstract pronominal element PRO [which is the subject of the infinitive in a sentence such as "I like to watch TV"]. (:5f)
>
> [B]inding and Case theory can be developed within the framework of government theory, and ... Case and θ-theory are closely interconnected. Certain notions, such as c-command, seem to be central to several of these theories. Furthermore, [these subsystems] interact: e.g., bounding theory holds of the rule Move-α (ie., of antecedent-trace relations) but not of other antecedent-anaphor relations of binding and control theory. Each of [the subsystems] is based on principles with certain possibilities of parametric variation. Through the interaction of these systems, many properties of particular languages can be accounted for .... Ideally, we hope to find that complexes of properties differentiating otherwise similar languages are reducible to a single parameter, fixed in one or another way .... (:6)

Obviously, this is not a complete introduction to GB-theory, but it should suggest the nature of the constraining principles that are involved in GB-theory and its variants.

5.2.4. A detailed Passive rule is no longer necessary

As an example, consider the detailed Passive rule (5). In modern terms, the old rule is not a separate rule of grammar, but merely one subcase of Move NP. The details mentioned in the old rule result from the interaction of various principles.

Passivization can't apply with active verbs because an active verb assigns a θ-role to its subject. The θ-criterion forbids a position that receives a θ-role from being empty at D-structure, so recoverability of deletion will prevent the object from moving into subject position.

Passivization must leave an empty trace behind because all movement rules do. It isn't necessary to stipulate that fact as a property of Move NP. (If there were no trace, a moved NP would lose its θ-role and violate the θ-criterion.)

Passivization is obligatory with passive verbs because of a principle of Case Theory that requires a noun phrase to have some case such as nominative or objective assigned to it. According to modern theory, passive participles do not assign case. The noun phrase in object position must move to a case-assigning position.

Various other details also follow. The copula *be* is required with passive verb phrases because they are thought to have the categorial status not of an ordinary verb phrase, but of a neutralized category intermediate between verb phrase and adjective. Subjacency, the major principle of bounding theory, rules out some other improper movements.

5.3. Factoring constraints out of grammars

The shift from detailed rules to systems of principles has also strengthened the theory of universal grammar. In addition to factoring constraints out of individual rules, syntactic theories can factor some constraints out of grammars entirely. Many constraints are thought

to hold for all natural languages and hence do not need to be stated in the descriptions of individual languages such as English and French.

When the properties of universal grammar have been factored out, the specification of the "core syntactic structure" of a language amounts to no more than a selection of particular values for parameters from a small list:

> Universal grammar will provide a finite set of parameters, each with a finite number of values, apart from the trivial matter of the morpheme or word list, which must surely be learned by direct exposure for the most part. (Chomsky, 1981:11)

These parameter settings interact with various principles to yield the language-particular effects that were attributed in earlier theories to the operation of detailed language-particular rules:

> Languages may select from among the devices of universal grammar, setting the parameters in one or another way, to provide for such general processes as those that were considered to be specific rules in earlier work. At the same time, phenomena that appear to be related may prove to arise from the interaction of several components, some shared, accounting for the similarity. The full range of properties of some construction may often result from interaction of several components, its apparent complexity reducible to simple principles of separate subsystems. This modular character of grammar will be repeatedly illustrated .... (:7)

In effect, recent theories can derive from deeper principles many syntactic facts that were merely written down (in the form of rules) in previous theories.

In linguistics, a theory of universal grammar that allows for only limited, parametric variation in basic structure from one language to another has three major advantages over one that allows a wide variety of complex, language-specific rule systems. It is preferred because of three major advantages. First, it makes stronger claims about the nature of natural languages, hence is preferred (if true) on general scientific grounds. Second, it limits the amount of information that is needed to characterize the structure of a language, hence can help make it possible to explain how a language can be acquired by children on the basis of limited evidence. Third, in cases where it can derive details of "rules" from general principles, it provides a better explanation for those details than a theory that simply writes them down.

5.4. Correcting the deficiencies of complex rule systems

§§5.2 and 5.3 suggest that by untangling the effects of separate underlying subsystems of grammar, modern theories of grammar have come closer to uncovering the principles that form the true basis of syntactic knowledge. The new theories postulate simple, restricted rules instead of complicated ones drawn from an unrestricted framework. They view syntactic variation from language to language as characterized by a small number of parameters rather than a large body of detailed rules. Where possible, they have factored out general conditions both from rules and from language descriptions.

The development of modular theories has cured many of the scientific ills of earlier theories of grammar. Fundamental principles support better explanations than stipulative rules. Separating different grammatical components from one another allows stronger constraints and more sweeping simplifications within each component. The theory of limited parametric variation condenses language descriptions to a small size and makes language acquisition seem possible.

6. Replicating the shift to modular syntax

Old-style grammatical theories and current natural-language parsers both use complicated rule systems to describe the syntactic structures of languages. Such rule systems have been superseded in grammatical theory because of scientific shortcomings, and for corresponding reasons they cause difficulties in parser design as well.

The cure for these engineering maladies should be the same in parser design as it was in linguistic theory. The shift to modular theories of syntax should be replicated in parsing practice. Such a shift would make it easier to design natural-language systems because it will shorten and simplify the necessary underlying descriptions of particular languages.

6.1. Rules and principles in parsing

A successful branch of linguistic theory has largely abandoned complicated, language-specific rule systems in favor of simpler subsystems of principles that can account for many of the same facts. Given the engineering disadvantages of old-style rule systems, why hasn't parser design already followed suit? The answer lies partly in the fact that there are no well-understood ways of using the new modular linguistic theories concretely in the processing of sentences.

It is fairly clear how to embed a context-free grammar in a parser; many parsing methods for such grammars have been developed. More generally, it is often easy to imagine many ways to base a parser on a system of rules that is explicit about such matters as the surface order and composition of the constituents of various constructions. In many cases the rules can be put to use in relatively direct fashion for the recovery of syntactic structure.

For example, an $SLR(0)$ parser (Aho and Ullman, 1977) can be said to use context-free grammar rules rather directly because it operates by simply tracing through the grammar rules, placing dots in the rules to indicate its position. Since a context-free grammar explicitly spells out the order and type of constituents in various constructions, it is a simple matter to keep track of what is expected next:

- If the item $A \Longrightarrow B.aC$ is currently one of the possible descriptions of progress so far through the input, this means that a phrase of type A is expected and its first constituent, a phrase of type B, has already been processed. An a can be expected next; if the next input symbol that is read is indeed an a, the item is advanced to read $A \Longrightarrow Ba.C$.
- When an item with a dot at the end becomes current, it means that the end of an expected constituent has been reached; if $A \Longrightarrow BaC.$ is a current item, then any

item of the form $P \Longrightarrow Q.AR$ that was current before the A-phrase was sought should now be advanced to read $P \Longrightarrow QA.R$.

- When an item with a dot before a phrase symbol such as A becomes current, it means that a phrase of the indicated type is expected next. The rules expanding that phrase type are consulted to start the parser off on its path through the expected phrase; when $P \Longrightarrow Q.AR$ was first current, the expected A-phrase would have been sought through activation of the item $A \Longrightarrow .BaC$.

Although I have suppressed many details in this description of $SLR(0)$ parser operation, it should be clear that the $SLR(0)$ parsing method makes direct use of the information about constituent order and constituent type that is spelled out in context-free grammar rules.

It is less clear how to implement a parser for a linguistic theory in which constituent order and constituent type in a construction are not explicitly spelled out, but follow instead from the interaction of various general principles and requirements. Such a theory does not directly say what various constructions look like on the surface; indeed, it would be redundant for the theory to do so, since it derives surface characteristics from other principles. As a consequence, it is more difficult to see how the parser can bridge the gap between structure and surface appearance.[9] Few implementation models are known for the new modular, principle-based theories of grammar. The models that do exist use the principles of grammar only indirectly.

Berwick and Weinberg (1984), for instance, point out that the Marcus parser can be considered an implementation of a recent linguistic theory because it uses similar representations and mimics similar constraints. However, the rules and organization of the Marcus parser do not correspond directly to those proposed by theorists. "Metarule" systems such as that of Gazdar (1981) also can also be used to implement new-style transformational theories.[10] However, the function of metarules is the precomputation of a large set of ordinary context-free rules. It is the context-free rules rather than the underlying grammatical principles that are then put to use in sentence processing. A metarule implementation of a new-style theory destroys its modular character by multiplying out the surface consequences of its various components. The context-free "object grammar" that results from applying metarules to a context-free base can be quite huge — containing "literally trillions of rules," in the words of Shieber (1983:4).

6.2. A research proposal

Computational linguistics should fill this gap in our understanding of how to put linguistic knowledge to use in sentence processing. Researchers should replicate in parsing practice the shift to modular, principle-based theories of syntax. According to recent linguistic theory, complicated, language-specific rule systems do not form an important part of a person's syntactic knowledge. Perhaps, then, such systems need not form the basis for the recovery of syntactic structure.

The research program that is proposed here seeks to discover how to base a parser on

[9] Whatever effect that fact has on the difficulty of parser design, however, it does not imply that parsing will be "less efficient" than with a surface-oriented system. That question could come out either way.

[10] This is not the interpretation that proponents of metarule systems intend.

interacting principles and parameters rather than on rules that individually stipulate the details of their operation. It should be possible to use many of the principles "directly" in parsing, without precomputing their effects. If parser operation were based on linguistic principles rather than large sets of stipulations, results from universal grammar could shorten and simplify the language descriptions used in natural-language parsing. Parser design as well as linguistic theory would be able to view syntactic variation from language to language as characterized by a small number of parameters rather than a large body of detailed rules. The notion of relatively direct realization of a grammatical theory could also be clarified.

More concretely, a parser could use principles rather than stipulative rules to detect the site of NP-movement in a passive sentence. It would insert a trace after the passive participle not because the grammar writer had written a language-particular rule that explicitly directed it to do so, but because it in some way directly respected the principles of case and θ-role assignment that force the conclusion that the post-participial position must have been a movement site.

6.3. Characterizing the proposed research program

The research program that is proposed here should draw on methods and results in several intellectual disciplines:

- It is a problem in *applied computer science* to investigate appropriate implementations of linguistic theories. In computer science, an abstract object is characterized by the set of operations defined on it. A description of the representations, principles, and rules that a linguistic theory postulates can serve as the specification for a family of abstract objects that help implement a parsing model for the theory.
- It is a problem in *applied linguistic theory* to take an account of the speaker's knowledge of language and put it to use in recovering the syntactic structure of sentences.
- It is a problem in *engineering* to try to improve the performance of natural-language processing systems. A parser that is based on an accurate and explanatory theory of linguistic structure has a better chance of accurately recovering that structure than a parser that is based on a large set of complicated rules. (There is no advantage, however, unless the implementation is both faithful and computationally practical.)
- It is a traditional goal in *artificial intelligence* to work toward systems that can learn rules instead of having them all built in. Learnability is an explicit concern in modern generative linguistics, and there is more hope of setting from experience the values of tightly constrained linguistic parameters than of inductively building up a complex set of rules.
- It expands the realm of *parsing theory* to propose new parsing algorithms. As noted, most current parsers are driven by sets of rules that directly specify constituent order. In contrast, the proposed new parser is to be driven by sets of principles that indirectly determine constituent order.
- It is of interest in *cognitive psychology* to propose new models of how knowledge of

language can be put to use in sentence processing. Each new parsing algorithm is potentially a new candidate for a model of how humans process language.

- It is of some interest in *linguistics* to discover in what respects a theory does and does not suffice to determine the structure of sentences. A parser implementation that actually processes sentences cannot help but shed light on this question.

The entire project amounts to taking modern linguistic theory seriously, toward a variety of ends.

6.4. Encouraging anecdotes

The ultimate success of this line of research cannot be predicted ahead of time. However, there is anecdotal evidence that the effort to factor out underlying principles instead of describing their surface effects can in fact yield engineering benefits as expected. Small is beautiful when it comes to the amount of information a parser designer must specify in order to parse a new language, and these examples show that a shift toward modular organization can indeed decrease the size and complexity of a parsing system. When independent but interacting underlying principles are involved, a rule system that multiplies out their surface effects is clumsy.[11]

6.4.1. Factoring out Aux-Inversion makes Marcus's question rules simple

Redundancy in a rule system often signals that the process of factoring surface appearances into underlying principles is not complete. The ability of modular factoring to reduce redundancy is illustrated nicely by the contrast between Robinson's and Marcus's treatments of yes/no questions. Consider the rules for yes/no questions in the DIAGRAM parsing system of Robinson (1982):

(21)
```
SQ = BEP NP ((ADV) (ING "BE") PRED)
SQ = MODALP NP (ADV) PPL "BE" ((ING "BE") PRED)
SQ = HAVEP NP (ADV) PPL "BE" ((ING "BE") PRED)
SQ = DOP NP (ADV) VP
SQ = MODALP NP (ADV) (HAVEP PPL) (BEP ING) VP
SQ = BEP NP (NOT) ING VP
SQ = BEP "THERE" (NP) ([ING ["BE" PRED1 / VP] / PRED2 / SREL])
SQ = (MODALP "THERE" (NOT) (HAVEP PPL) BEP
     (NP) ([ING ["BE" PRED1 / VP] / PRED2 / SREL])
SQ = HAVEP "THERE" PPL "BE" (NP)
     ([ING ["BE" PRED1 / VP] / PRED2 / SREL])
```

There is much redundancy in the statement of question rules themselves, and there is even more when a few of the completely separate rules involved in the corresponding declaratives are considered:

[11] The complexity that results from using a non-modular, surface-oriented framework reaches a striking extreme in the parsing system described by Sager (1981). Based on Zellig Harris's structuralist string-analysis framework and twenty years in the making, the system uses a plethora of rules and category types. It uses at least 100 expansions for the supposed category "object."

(22)
```
SDEC = "THERE" (AUX) BEP NP ([ING [VP / "BE" PRED1] /
                                PRED2 / SREL])
SDEC = NP (AUX) (ADV1) BEP (ADV2) (PRED)
AUX = (MODALP) (HAVEP PPL) (BEP ING)
SDEC = NP (ADV) (AUXD) VP
AUXD = [AUX / DOP]
```

Robinson (1982:32) seems aware of some aspects of the redundancy problem:

> [W]e feel that there is some loss of generality in writing so many separate rules that have so many elements in common, and we are therefore exploring the possibility of deriving some rules from other rules.

In contrast, Marcus (1980) is not constrained by his rule framework to spell out surface configurations, and he is able to capture many of the consequences of Robinson's complex rule set for questions by supplementing his independently required rules for actives with two simple parsing rules that are related to the traditional transformation of Aux-Inversion:

(23)
```
{rule YES-NO-Q in ss-start
 [=auxverb] [=np] -->
 Label c s, quest, ynquest, major.
 Deactivate ss-start.  Activate parse-subj.}

{rule AUX-INVERSION in parse-subj
 [=auxverb] [=np] -->
 Attach 2nd to c as np.
 Deactivate parse-subj.  Activate parse-aux.}
```

In the Marcus parser, as in transformational grammar, surface constituent order in yes/no questions is taken to be derivative. The surface order is not to be spelled out directly with non-modular rules like those in Robinson's grammar. Rather, the word order of yes/no questions results when the processes that determine word order in declaratives are perturbed by a separate process that Marcus factors out as a simple pair of rules.

6.4.2. Separating syntax from semantics also simplifies rule systems

The complexity of a parsing system can also be reduced when knowledge of linguistic form and knowledge of the world are separated into different modules. On the theoretical side, Grimshaw (1979) showed that mixing syntactic and semantic requirements led to a larger overall description of the relationship between verbs and their complements. On the practical side, mixing the semantic component of a program into its syntactic rule system will typically make it necessary to duplicate the same semantic tests and actions in the rules that parse all the syntactically distinct ways of expressing roughly the same idea. Robinson (1982) mentions the undesirability of duplicating a single semantic action in several different grammar rules, and the experience of Bates with semantico-syntactic and purely syntactic ATN systems also seems to support this point:

> Semantic grammars tend to be much larger than syntactic grammars which accept the same set of sentences. The largest ATN grammar this

> author knows of is one she wrote for the BBN speech understanding system ...; it contained 448 states, 881 arcs, and 2280 actions but was more limited in the variety of constructions it could accept than an 83 state, 202 arc, 386 action syntactic grammar for the same system. (Bates, 1978:238)

Another practical advantage of keeping syntax separate from semantics is that — once again — it saves the parser designer from the task of explicitly working out the interactions and writing them into the rules. Bates continues:

> Another drawback to a semantic grammar is that it must be written anew if the domain of discourse is changed, and it would be extremely impractical to attempt to write such a grammar for anything but a limited application area. (:238)

Evidently a modular approach leads to a system that is both less bulky and easier to modify.

7. Some possible characteristics of the proposed parser

This section suggests some tentative choices for the design of the parser. Some choices must be made in order to preserve the benefits of modular syntactic theories. Others are not forced and thus represent only one point in the spectrum of possible research strategies.

7.1. Parsing under the control of principles of grammar

The general function of a parser for a grammar G is to "understand sentences in the manner of G."[12] The parser carries out this function by assigning structural descriptions "under the control" of principles of grammar. This characterization of parser operation leaves open a spectrum of options for the degree of directness of "control" by the grammar. At one extreme of the spectrum, control by the grammar might amount only to the imposition of an input/output constraint.[13] The operation of the parser would then be determined largely by performance principles distinct from rules of grammar. At the opposite extreme, the grammar might force each internal step of parser operation. The contribution of performance principles would then be much smaller.

The spectrum of directness of control corresponds to a related spectrum of directness of constraint on internal representations. At one end of this spectrum, the requirement that the parser "understand sentences in the manner of G" might be enforced only at parser output. The operations of the parser would be allowed to build internal representations that did not satisfy the principles of G so long as they did not surface at the output. At the opposite end, the requirement might be enforced incrementally as invariant constraints on internal representations. The structure-building operations of the parser would be constrained so as to build representations satisfying the principles of G. Satisfaction of the output constraint

[12] This phrase is from Miller and Chomsky (1963).

[13] The impractical "British museum algorithm" of generating all possible structural descriptions and using the grammar to rule out those that are unsatisfactory falls close to this end of the spectrum, but the kind of "unnatural" parsing algorithm described by Aho and Ullman (1972:272) falls even closer, since it does not use the grammar at all, but only happens to produce the right derivations.

would follow as a special case.

Control and constraint of a parser by the principles of universal and particular grammar can thus be direct or indirect. The goal of translating into parser design the benefits of modern principle-based linguistic theories suggests that a relatively direct implementation might be appropriate. One can envision a parser that is a direct implementation of GB-theory (Chomsky, 1981) in that it recovers the linguistic descriptions that GB-theory proposes, it recovers them by actively using the principles that GB-theory holds to characterize and define them, and it uses those principles without "multiplying out" their effects to produce an intermediate set of phrase-structure rules. In terms of Fodor, Bever, and Garrett's (1974) typology of parsing methods, it would probably use elements of both analysis by analysis and analysis by synthesis.

7.2. Preserving the structure of grammatical theory

A certain amount of directness is required in an implementation that strives to retain the benefits of a modular grammatical theory:

- The distinction between language-independent *universals* and language-particular *parameters* must be preserved if descriptions of individual languages are to remain small.
- Other aspects of the *modular structure* of grammatical theory must be preserved if the combinatorial consequences of multiplying out interaction effects are to be avoided.
- The set of *linguistically significant operations* should be preserved in parsing representations so that the parser designer cannot accidentally write rules that refer to predicates that are linguistically nonsignificant. For example, the parsing framework should not lead the grammar rules to place much more importance on the *left* versus *right* distinction or the notion of *string position* than is warranted by the syntax of natural languages.
- It is desirable for the structure of *explanations* to be mapped over intact from grammatical theory to parser operation. If certain grammatical principles force the assignment of a certain structure to a sentence, corresponding implementation principles should be responsible for the parser's decision to assign that structure.

However, see section 8.1.3 for some limits on the degree of directness that it is reasonable to impose. Making the notion of direct implementation more precise should be a subsidiary topic of research in this program.

7.3. Avoiding mysteries

One reason for investigating the possibility of a relatively direct relationship between grammar and parser is that indirect relationships, though not logically impossible or yet empirically falsified, nonetheless give rise to what might be regarded as mysteries.

Consider, for example, a metarule account of grammar. On such an account, the properties of a language can be specified with a small set of context-free base rules plus a

set of metarules to derive new rules from old ones. The parser, however, operates only with the large set of derived rules.

It is possible to imagine that the human language faculty could be constituted roughly along the lines of the metarule account. The human parsing mechanism would then be capable of using an unrestricted set of context-free rules for parsing, but languages described by "unnatural" sets of context-free rules would never be observed because the language-acquisition component of the language faculty would never construct such a set. Applying only at the level of language acquisition, the constraints of universal grammar would play no role in actual parser operation.

In a way, however, any account that involves translation of restrictive principles into a less restricted framework reads like a mystery story. If the language-acquisition component has the option of using powerful computational devices in the rule systems that it constructs, it seems a mystery why it never uses them. If only a limited range of the parser's computational abilities are ever needed, it seems a mystery why the parser isn't tailored to take advantage of the restrictions on its actual computational problem. Of course, the actual nature of the parser and of the language-acquisition component are matters for empirical investigation. It might turn out that the correct theory of the human language faculty is one that seems at first to have mysterious properties. Nevertheless, the attempt to avoid mysteries is a sufficient reason for trying to investigate the "direct" rather than the "indirect" line of parser implementations.

Related qualms come to mind about the "parsing strategies" proposed by Fodor, Bever, and Garrett (1974). If the parameters and principles of grammar are not directly realized in the parser, one must ask for an explanation of why the structures that the parser recovers are in accord with those principles.[14] Such a question seems to arise whenever a system *observes* principles that play no *causal role* in its operation.

7.4. Separating competence and performance principles

In the design of a parser it is desirable to preserve as much as possible the distinction between competence and performance principles. If performance instructions must be explicitly written into the language description that the designer of a natural-language system must write, language descriptions will remain large and complex. In addition to specifying the parameters that characterize the core syntactic structure of a language, the system designer will be forced to describe in the rule system the detailed way in which those parameters relate to surface evidence. As much as possible, the necessary performance principles should be applied automatically by the parsing machinery.

Many complicated rules in Marcus's grammar seem to be concerned with questions like these:

- If an optional constituent *isn't* attached at the current level, will some higher syntactic context be able to receive it?
- If an optional constituent it is attached at the current level, will some higher syntactic context be deprived of an obligatory constituent?

[14] There may be an answer, of course; it is again an empirical question whether realization of parameters is direct.

- If the *wh*-comp *is* used at the current level, will some higher syntactic context be able to receive the NP that is rejected at this level and left over?
- If the *wh*-comp *isn't* used at the current level, will it be possible to use it elsewhere?

These considerations seem amenable to incorporation into the structure of the grammar interpreter so that they will not need to be explicitly written into grammar rules. If the grammar interpreter could handle most performance principles systematically and automatically, without needing explicitly coded rule actions, the burden on the grammar writer could be reduced. It might become less of a misnomer to call Marcus's rule sets grammars, if rules came to encode more knowledge of the structure of a language and less about the details of recovering grammatical structures from local cues available from input strings.

Marcus's diagnostic rules are a case in point. It would be easier to write language descriptions for Marcus's grammar interpreter if the intepreter were made explicitly aware of the process of resolving nondeterminism. With Marcus's original grammar interpreter, nondeterminism is resolved when the grammar writer notices a rule conflict or an overly general interpretation of a surface cue and then writes a diagnostic rule to distinguish between two situations by using semantics, examining the active node stack, or using more lookahead. The grammar interpreter itself treats diagnostic rules like any other rules; it does not "know" when it has gotten into trouble and should consult some mechanism to resolve a conflict.

The grammar writer therefore has the burden of foreseeing and resolving surface ambiguities and interaction effects. This increases the size and complexity of the rule system, and in practice a few techniques for resolving nondeterminism seem to be repeated over and over in different grammar rules. If the grammar interpreter recognized conflicts and invoked explicit resolution procedures, the modularity of the parsing system could be improved because fewer grammar rules would explicitly perform exotic tests.

7.5. Logical parsing theory

One possible strategy for designing a parser involves studying *what* surface cues to syntactic structure are available in an input sentence before deciding *how* to use those cues in guiding the recovery of structure. The possible theoretical level of "logical parsing theory" would concentrate more than grammatical theory on the nature of the computational problem that the parser must solve, but it would leave open the question of how the parser actually goes about using (some or all of) the information in the surface string. In Marr's framework (§2.2), both grammatical theory and logical parsing theory are part of the topmost level of *computational theory,* which both identifies the goal of the computation and investigates the logic of the strategies by which it may be carried out. Once logical parsing theory is available, it becomes a computer science problem to devise a detailed parsing algorithm using some subset of the available structurally relevant information.[15]

Given a formalization of crucial principles from linguistic theory, it should be possible to derive theorems about the surface appearance of underlying constructions. These theorems

[15]The results of an experiment by Frazier, Clifton, and Randall (1983) suggest that the human parsing mechanism does not apply all potentially relevant constraints while computing a structural description. Some constraints are apparently not applied until a later stage.

can then be put to use in parsing. Some theorems will be ultimately based only on linguistic universals, without mention of language-specific parameters. These theorems can be used, where appropriate, to fix the general structure of the parser. For example, bounding theory can influence the choice of parser architecture because it places limits on the "range of search" that the architecture must support. In general, the relationship between a universal principle and its embodiment can be quite indirect without raising questions of language acquisition and language-description size. If fundamental parameters of cross-linguistic variation are embodied in highly indirect fashion, however — if, for example, they show up as variations in the architecture of the parser — it will be necessary to consider closely how they might be set from experience in language acquisition or concisely stated in parser design.

One possible topic for logical parsing theory is the appropriateness of various internal representations that a parser might use. Given a sufficiently developed linguistic theory, it is possible to investigate how closely a proposed "representation cluster"[16] for the theoretical objects postulated in the theory conforms to the goal of displaying in a representation all and only the information that is grammatically relevant according to the theory. How should key notions such as dominance, adjacency, government, c-command, projection, subjacency, and the contiguity of constituents be reflected in the parser's representations? Given a representation and a set of operations, can all S-structures be derived by means of the operations?

7.6. A target for parser coverage

The enterprise of constructing a parser always involves decisions about the range of constructions that are to be correctly processed. Given the nature and purpose of the research program that is proposed here, a reasonable target would be to handle all syntactic constructions of "core grammar" that are regarded as fundamental and reasonably well-understood in discussions of GB-theory. This does not include all of language:

> [I]t is hardly to be expected that what are called "languages" or "dialects" or even "idiolects" will conform precisely or perhaps even very closely to the systems determined by fixing the parameters of universal grammar. This could only happen under idealized conditions that are never realized in fact in the real world of heterogeneous speech communities. Furthermore, each actual "language" will incorporate a periphery of borrowings, historical residues, inventions, and so on, which we can hardly expect to — and indeed would not want to — incorporate within a principled theory of universal grammar. For such reasons as these, it is reasonable to suppose that universal grammar determines a set of core grammars and that what is actually represented in the mind of an individual even under the idealization to a homogeneous speech community would be a core grammar with a periphery of marked elements and constructions.
>
> Viewed against the reality of what a particular person may have inside his head, core grammar is an idealization. From another point of view, what a

[16] This is a term from Liskov (1977).

> particular person has inside his head is an artifact resulting from the interplay of many idiosyncratic factors, as contrasted with the more significant reality of universal grammar (an element of shared biological endowment) and core grammar (one of the systems derived by fixing the parameters of universal grammar in one of the permitted ways). (Chomsky, 1981:8)

Core notions and mechanisms are, however, expected to play a role in determining the properties of even peripheral constructions:

> We would expect the individually-represented artifact to depart from core grammar in two basic respects: (1) because of the heterogeneous character of actual experience in real speech communities; (2) because of the distinction between core and periphery. The two respects are related, but distinguishable. Putting aside the first factor — *i.e.*, assuming the idealization to a homogeneous speech community — outside the domain of core grammar we do not expect to find chaos. Marked structures have to be learned on the basis of slender evidence too, so there should be further structure to the system outside of core grammar. We might expect that the structure of these further systems relates to the theory of core grammar by such devices as relaxing certain conditions of core grammar, processes of analogy in some sense to be made precise, and so on .... (:8)

Even so, it would be premature to tackle the periphery without first devising an implementation that can handle the core.

In addition to foregoing treatment of peripheral constructions in language, initial stages of the proposed endeavor should avoid getting mired in several other issues: functional explanations for linguistic phenomena, the "communicative function of language," "everyday language," "situated language," and the matters that might be called "semantic issues" in the broad sense. If tackled too early, these issues cannot fail to impede progress toward the development of a principle-based parser.

8. Implicit representation

Section 7.5 raised the question of how grammatically relevant predicates and conditions might be represented in the proposed parser. One possible answer involves implicitly embodying some predicates and conditions in the structure of the parser. This preliminary section explores that possibility.

8.1. Implicit representation

Under the proposed line of parser development, it is usually not enough for the parser to *observe* the principles of grammatical theory. Rather, the principles are to play a relatively direct role in determining what action the parser should take at each point. In most cases the principles are to be causally implicated in explanations of parser behavior: it is to be the principles, and as little else as possible, that are actively used to determine the structural description to be assigned to an input.

8.1.1. The implementation must preserve the benefits of modularity

More specifically, however, the exact degree of implementation "directness" to be achieved is uncertain. Indeed, much remains to be explicated about notions such as "direct implementation" and "direct use" of principles, and such explication will be part of the thesis project. Nonetheless, the main constraint on the directness of the proposed parser implementation is clear: the implementation should preserve enough of the modularity of modern grammatical theory so that the possible benefits of that modularity will not be lost.

In part, then, the reason why the implementation is to be direct is so that the shift to modular syntactic theories can reduce the size and complexity of language descriptions in parser design just as it did in linguistic theory. Given that goal, the most important modular division to preserve in the mapping from theory to program is the distinction between language-particular parameters and linguistic universals.

Consider, for example, a parser that produces the syntactic descriptions demanded by current syntactic theories but whose operation is based on a surface-oriented rule system. Such a rule system must "multiply out" the surface effects of various principles and parameters. How much will such a rule system differ from the rule system for a closely related language? Recall the effects of changing a single parameter in modern grammatical theories:

> In a tightly integrated theory with fairly rich internal structure, change in a single parameter may have complex effects, with proliferating consequences in various parts of the grammar. Ideally, we hope that complexes of properties differentiating otherwise similar languages are reducible to a single parameter, fixed in one or another way. (Chomsky, 1981:6)

In accordance with this picture, a single surface-oriented rule expresses not a fundamental parameter of linguistic variation, but a complex amalgam of different principles and parameters. Changing a parameter causes large changes in the rule system. The benefits of modularity are not obtained, for the language descriptions that the system designer must construct are large and vary greatly from language to language.

8.1.2. The implementation must often preserve the structure of explanations

One characteristic of explanations is that they support counterfactuals. By reducing situations to their underlying causes, an explanatory theory describes not only what *is* true, but also what *would be* true under a range of different conditions. By identifying the principles and parameters on which a language-particular phenomenon depends, an explanatory theory reveals not only why it has its particular characteristics, but how those characteristics would differ given a different set of underlying parameters.

An interesting corollary of this fact is that preserving the parametric structure of cross-linguistic variation requires the implementation to preserve the logical structure of some linguistic explanations. Moving from one language to another amounts to using a different set of parameters. Generally speaking, the implementation cannot take advantage of "lemmas" that are derived from grammatical principles and parameters. It should use the principles and parameters directly, because if it uses a language-particular lemma, the lemma will have to be stated in the rule system that makes up the description of the

language. A different lemma will have to be stated for a language with different underlying parameters; language descriptions will thus include more than the logically necessary sets of parameter values.

8.1.3. Implicit representation is often permitted and desirable

There are three major cases in which this argument does not hold, however. First, it does not apply to language-independent lemmas that do not depend on particular parameter values. Second, it does not apply to certain parametrized lemmas that show how parametric differences would affect their conclusions. Third, it does not apply when the parser rather than the system designer is responsible for generating the lemmas; in such cases the parser is only using the lemmas for "short-cut" access to results it could have derived from fundamental principles.

The first two cases allow the possibility of *implicit representation* of grammatical principles and predicates. These cases represent a situation in which it *is* acceptable for the parser to merely act in accordance with principles instead of basing its decisions on them directly. Language-independent lemmas can be used to fashion the basic architecture and actions of the parser. In fact, within the constraint of preserving parametric structure, it is *desirable* to tailor parser design closely to the general computational problem defined by universal grammar. Without close tailoring, it becomes a mystery (§7.3) why no language uses the full power of the implemented parser. The suspicion arises that some property of universal grammar has been missed that would allow a more efficient or otherwise more desirable parser architecture.

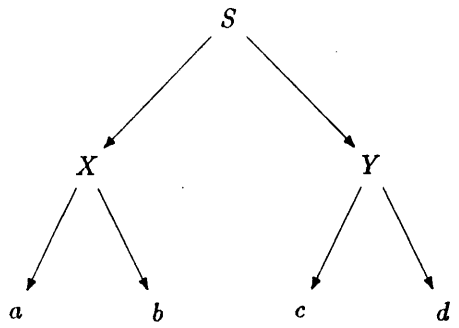
8.2. Monostrings

Tailoring a representation to closely fit a desired set of predicates, operations, and constraints is nothing new to linguistic theory. Indeed, the search for appropriate representations is a key part of the construction of theories that can explain language acquisition. As the theories go, the language learner projects from limited experience in one way rather than another because the human language faculty makes available only a restricted framework for describing linguistic experience. Children never frame arbitrarily bizarre hypotheses about the structures of their languages because such hypotheses are not statable with the available internal vocabulary.

8.2.1. The monostring representation implicitly embodies universal restrictions

Lasnik and Kupin (1977) have described a restricted transformational framework that provides an example of the attempt to build the restrictions of universal grammar directly into the formalism used for stating rules of grammar:

> The theory differs from [Chomsky's earliest transformational formalism], and more markedly from most current theories, in the extent to which restrictions are imposed on descriptive power. Many well-justified and linguistically significant limitations on structural description and structural change are embodied in the present formalism .... In this paper



$\{abcd, S, Xcd, abY\}$

Figure 6: In the framework of Lasnik and Kupin (1977), the tree on the left could be described by the set of strings on the right. Except for the terminal string, each string in the set is a monostring.

> we are attempting to present a particular theory of syntax in a precise way. Many of the operations describable within other theories cannot be expressed within this theory. (:173)

Lasnik and Kupin's restrictive formalism helps illuminate the possible nature of the representations used by the human language faculty. Many properties of the set of possible transformations would follow from the assumption that the language faculty uses a representation like the one that Lasnik and Kupin propose.

Lasnik and Kupin use so-called *monostrings* to capture the hierarchical relationships that are usually represented with tree diagrams. Each monostring represents a particular occurrence of a phrasal category. With a monostring representation, a phrase-marker is a set of strings instead of a tree. Figure 6 gives the monostring representation that corresponds to a simple tree.

The monostring representation is more closely tailored to certain theories of universal grammar than a tree representation would be. It fails to represent certain distinctions that a tree would represent, and therefore it is suited only to a theory of universal grammar in which those distinctions are never relevant for the description of natural languages. Figure 7 shows two trees that have the same monostring representation. Lasnik and Kupin comment:

> The choice of [the monostring representation], then, constitutes an empirical claim about human language. All grammars in this theory will necessarily treat [the two trees shown in the figure] identically since they have identical representations .... (1977:178)

The "pruning" of two identical nodes dominating the same material also follows from the nature of the monostring representation. A nonbranching node of the same type as its daughter is "invisible" with that representation. Again an empirical claim is made:

> [A reduced phrase-marker] is essentially a collection of *is a* statements. An *is a* statement concerns only the relationship between a portion of the ter-

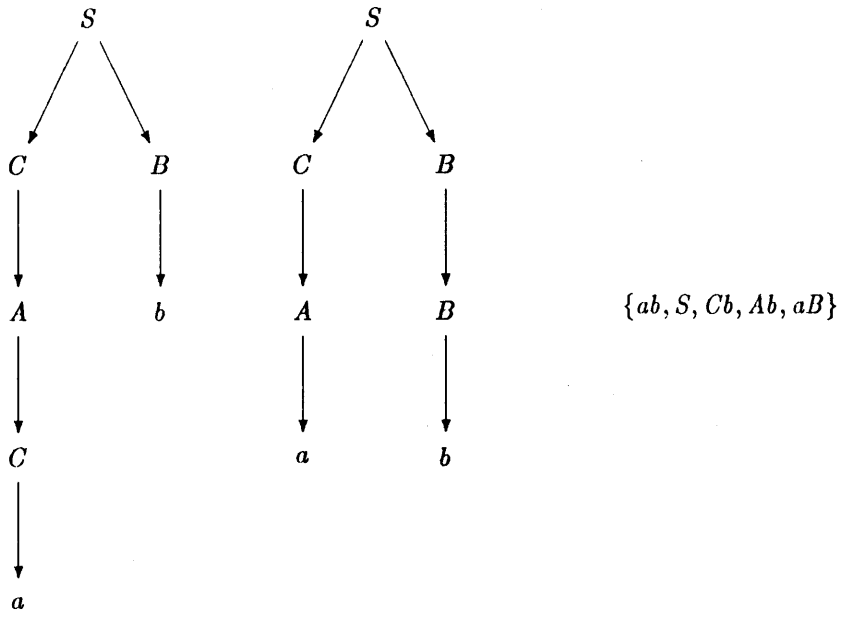


Figure 7: The two trees on the left have the same reduced phrase-marker, shown on the right. This example is taken from Lasnik and Kupin (1977).

> minal string and a non-terminal. In that view there is no point in saying a particular occurrence of a terminal [stands in the *is a* relationship to some other nonterminal] twice (as [the unpruned tree] apparently does) .... In this theory, pruning thus becomes a non-issue, since the repeated nodes never exist to be pruned. There is never a conversion to more tree-like objects so the issue never comes up. Thus, the effects of pruning, if indeed there are any, are unavoidable .... It is important to note than in principle a base component could distinguish between [the pruned and unpruned trees]. Thus, [by choosing this representation] we are making the claim that a transformational component does not require access to all of the information inherent in a base component. (:179)

The design of the proposed parser will strive for this kind of close fit between the information that is made explicit in the representation and the information that is deemed grammatically significant by linguistic theory. Mysteries arise when the representation displays a wide range of information that the grammatical system never uses.

Note that an implementation is not required to observe insignificant "presentation details" of linguistic theories. For example, some theorists who might actually prefer to use Lasnik and Kupin's framework still draw trees for expository convenience. Unlike some of the information they depict, the trees are not theoretically significant. In a mathematical sense, the trees are convenient *models* of linguistically significant statements.

8.2.2. Only a restricted class of transformations can be stated

Lasnik and Kupin's framework also restricts the class of transformations that can be stated. Again, the restrictions are intended to embody empirical assumptions:

> All of the definitions and all of the principles of application described below are assumed to be part of general linguistic theory, *i.e., to be biologically based* [emphasis added]. (1977:179)

A leftward NP-movement transformation would be stated as (24) in Lasnik and Kupin's framework:

(24) (NP NP, (2/1))

Explicit variables are not allowed in the statement of a transformation; there are implicit variables between all consecutive elements and hence no transformation can require two elements to be adjacent. There are no Boolean combinations of string conditions, clausemate conditions, or multiple analyzability conditions. Transformations are not marked optional or obligatory. There can be at most two affected constituents, so an NP-movement transformation cannot also insert a passive morpheme. There are only a finite number of possible transformations.

Again, this restrictive framework illustrates the kind of close match between theoretically permissible operations and representationally expressible operations that should be heavily used in the proposed parser design.

8.3. Subjacency

The subjacency constraint provides another natural opportunity to implicit represent grammatical constraints in the design of a parser. Subjacency is a constraint on movement that to first approximation forbids moving in one jump across more than one "bounding category," where NP and S are bounding categories.[17] For example, subjacency (so formulated) forbids *wh*-movement from applying to produce (25):

(25) *who do you believe $[_{NP}$ the claim $[_{S}$ that Bill saw e]]?

Stated another way, the subjacency constraint requires that movement transformations must apply to elements in the same domain or adjacent domains.

Marcus (1980, Chapter 6) attempted to show that important subcases of the subjacency constraint followed naturally from the structure of his grammar interpreter. In turn, the crucial properties of the grammar interpreter were motivated by its deterministic operation. For reasons that I will not detail here, Marcus's arguments do not completely go through. However, Berwick and Weinberg (1984) show that a constrained, deterministic parser of a certain kind must obey some principle similar to subjacency.

Fodor (1983) presents two possible treatments of phenomena related to subjacency. They contrast sharply with the general approach followed by Marcus and by Berwick and Weinberg. First, considering an ATN hold-cell parsing model, she proposes that "island constraints" (such as the Complex NP Constraint, derived from subjacency in many current

[17] Some modern theories of subjacency, such as the theory described in Chomsky's Fall 1983 class lectures, are more complex.

theories) can be handled by using a structured hold cell and arranging for the parser to follow "a set of traffic rules to control access to its various levels" (:172).

Second, working in a surface-oriented context-free framework, she suggests that the subjacency constraint, if needed at all, should be handled by arranging for the rule system not to include any rules that would violate it:

> Gazdar has no analog to Subjacency in his system at present, but at least for English a comparable effect could be achieved if passage of a slash annotation through either an S or an NP node was blocked, but a rule was added to allow a slash annotation at the top of a complement clause to be cashed out as a trace, and a new path of slashed nodes to be initiated with this trace as its filler. For Italian, however, Rizzi's analysis clearly requires that transmission of a slashed node be blocked only at the second of two cyclic nodes, and this entails ... [that] the slashed nodes on the path would have to be tagged with information about dominating cyclic nodes. (Fodor, 1983:191)

In both cases Fodor is effectively suggesting that the subjacency constraint should be descriptively imposed on top of the constraints (if any) that result from basic parser structure. The line of parser development that is suggested here rejects such descriptive approaches, which simply describe the effects of principles and conditions instead of building them fundamentally into the actions and representations of the parser. Given the desire to base parser design solidly on linguistic theory, it is better to avoid a parsing framework in which the parser would work just as well for parsing "unnatural" languages (with properties unattested in natural languages) as for parsing natural languages.

One can imagine several possible parsing approaches to the subjacency constraint. It is necessary to give some account of why subjacency treats certain domains as units; perhaps it will be possible to find some process necessary to other aspects of parser operation that already treats those domains (and only those) as units. A contrary approach is also possible; perhaps subjacency domains are not agglomerated as units, but instead the intervening nonbounding material is dismissed from some relevant local store and hence is not "visible" to hinder movement. It is desirable to explain why two domains can be involved, not three or just one.[18] Fodor (above) suggested the "barrier" account in which search-barriers are inserted into memory stores for some reason. Again a contrasting approach is possible; perhaps there is not barrier insertion, but rather the temporary dismissal of material that would be hidden by a barrier.

8.4. C-command

Berwick and Weinberg (1984, Chapter 5) provide a final example of implicit representation. They show how the structure of the Marcus parser can be adjusted so that the grammatically relevant predicate of *c-command* (Reinhart, 1976; Aoun and Sportiche, 1983) does not need to be explicitly computed, but is implicitly available as a by-product of normal

[18] See Berwick and Weinberg (1984) for one explanation, based on the observation that "grammars can't count." Another possibility might somehow involve a process in which the parser is trying to relate two partially built structures.

parser operation:

> While traces an nontraces diverge with respect to bounding conditions, traces and some of the nontrace categories are similar in that they both obey c-command. Traces, pronouns bound by some quantifiers, and lexical anaphors must be c-commanded by their antecedents.
>
> Given that c-command is a basic predicate of the government-binding theory, we must be able to compute it from the parsing representation. The obvious way to do this would be to use a full tree representation and then design an algorithm to compute c-command from it.
>
> Alternatively, we could build on the fly a representation that makes the calculation of c-command computationally trivial. This is the tack that we shall take. (Berwick and Weinberg, 1984:173)
>
> Given [a certain] principle of attachment and node completion, the active node stack extensionally represents the c-command predicate; c-command need not be separately computed. (:175)

This way of representing grammatically relevant predicates is quite attractive under the current proposal for parser design, since it represents very close fit between the set of grammatically relevant predicates and the design of the parser.

9. Relation to other work

As noted, the proposed research program builds on work in linguistics, computer science, psycholinguistics, and parser design. Chomsky (1981) describes the government-binding theory of grammar that is to be implemented. Lasnik and Saito (1983) provide recent revisions. Some variant of the monostring representation of Lasnik and Kupin (1977), which was discussed in section 8.2, could potentially form the basis for the parser's representation of hierarchical structure. Stowell (1981) discusses in detail many cases in which grammatical principles can interact to account for the surface constituent-order facts that were formerly accounted for with detailed phrase-structure rules. The proposed parser is to make crucial use of interacting principles to determine surface order.

Aho and Ullman (1972) discuss formal results about so-called *covering grammars* that could be useful in clarifying the notions "implementation of a grammatical theory" and "direct implementation." Liskov *et al.* (1977) and others have discussed and developed the notion "implementation of an abstract object." In the proposed research, linguistic theory will be taken to define a family of abstract objects that it is the parser's job to implement.

Fodor, Bever, and Garrett (1974) survey psycholinguistic results that may (with caution) be interpreted to describe some properties of human language-processing mechanisms. More recently, Frazier and Rayner (1982) and Frazier, Clifton, and Randall (1983) have done experiments that bear on the question of how the human syntactic processor deals with parsing ambiguities. Questions about such ambiguities were an important factor in the design of Marcus's (1980) deterministic parser and will probably also significantly influence

the proposed parser design. Works by Seidenberg *et al.* (1982) and Milne (1983) present further psycholinguistic discussions.

The work of Marcus (1980) as refined and discussed by Berwick and Weinberg (1984) could easily play an important role in the proposed research. Of the currently available parsing models, the modified Marcus model is probably the most closely tailored to the principles of grammatical theory. However, the model should be further modified so that it bases its operation on principles and parameters rather than a complex rule system. Milne's (1983) work is also in the Marcus framework.

The proposed parser would implement grammatical principles more directly than existing parsers in the Marcus framework. Berwick and Weinberg point out that the rules for their modified Marcus parser are several steps removed from grammatical principles; in effect, the rule system expresses derived lemmas as discussed in section 8.1. Works by Gazdar (1981) and Fodor (1974) may be useful for contrast, since in many ways they fall at the opposite end of the spectrum from the proposed parser. Bachenko *et al.* (1983), Wehrli (1983), and Shieber (1983) sketch parsing models that may be relevant.

10. A suggested plan for initial research

One possible plan for the construction of a principle-based parser begins with two complementary prongs of initial attack. On the one hand, the Marcus parser can be successively modified to base its operation on principles and parameters in more and more cases. For example, the first case to be tackled might be the NP-movement case mentioned in section 6.2. As this effort proceeds, a working set of concrete examples could be built up. A "map" of the logical relationships among various principles and parameters could also be developed. A catalog of linguistic principles, constructions, and examples could be compiled.

On the other hand, efforts should continue to find representations and operations that are closely tailored to the predicates, operations, and parameters of grammatical theory. This second effort would thus aim for the development of possible alternatives to the Marcus framework. While the first-prong approach seeks to modify an existing model of parsing decisions and actions, the second approach follows section 7.5 in separating the question of *what* representations and actions the parser might use from the question of *how* it should decide what action to take at each point. The development of logical parsing theory is part of this line of research.

The results of this initial phase of research could support the choice of a particular parser design for further development. With a general design chosen, research can focus on reducing the amount of information that must be specified for parsing a particular language. The ideal limit of such reduction would allow a language to be described for parsing by the same set of parameters that specified the characteristics of the language in linguistic theory. In the ideal limit, all performance principles and interaction effects would be handled by the parser rather than the grammar writer.

11. Bibliography

Aho, A. V., and J. D. Ullman (1972). *The Theory of Parsing, Translation, and Compiling.* Vol. 1: Parsing. Englewood Cliffs, N.J.: Prentice-Hall.

Aho, A. V., and J. D. Ullman (1977). *Principles of Compiler Design.* Reading, Mass.: Addison-Wesley.

Aoun, J., and D. Sportiche (1983). "On the Formal Theory of Government," *The Linguistic Review* 2: 211-236.

Bachenko, J., D. Hindle, and E. Fitzpatrick (1983). "Constraining a Deterministic Parser," to appear in *Proc. Nat'l. Conf. on Artificial Intelligence* (AAAI-83).

Baddeley, A. D. (1976). *The Psychology of Memory.* New York: Basic Books.

Baltin, M. R. (1982). "A Landing Site Theory of Movement Rules," *Linguistic Inquiry* 13:1, 1–38.

Bates, M. (1978). "The Theory and Practice of Augmented Transition Network Grammars," in L. Bolc, ed., *Natural Language Communication with Computers,* Lecture Notes in Computer Science 63, 191–254. New York: Springer-Verlag.

Berwick, R. C., and A. S. Weinberg (1984). *The Grammatical Basis of Linguistic Performance.* Cambridge, Mass.: M.I.T. Press.

Chomsky, N. (1965). *Aspects of the Theory of Syntax.* Cambridge, Mass.: M.I.T. Press.

Chomsky, N. (1976). "Conditions on Rules of Grammar," *Linguistic Analysis* 2:4, 303–351.

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

Earley, J. (1970). "An Efficient Context-Free Parsing Algorithm," *Comm. ACM* 14, 453–460.

Fiengo, R. (1977). "On Trace Theory." *Linguistic Inquiry* 8:1, 35–61.

Fodor, J. A., T. G. Bever, and M. F. Garrett (1974). *The Psychology of Language.* New York: McGraw-Hill.

Fodor, J. D. (1983). "Phrase Structure Parsing and the Island Constraints," *Linguistics and Philosophy* 6, 163–223.

Frazier, L, and K. Rayner (1982). "Making and Correcting Errors during Sentence Comprehension: Eye Movements in the Analysis of Structurally Ambiguous Sentences," *Cognitive Psychology* 14, 178–210.

Frazier, L., C. Clifton, and J. Randall (1983). "Filling Gaps: Decision principles and structure in sentence comprehension," *Cognition* 13, 187–222.

Gazdar, G. (1981). "Unbounded Dependencies and Coordinate Structure," *Linguistic Inquiry* 12:2, 155–184.

Greenberg, J. H. (1963). "Some universals of grammar with particular reference to the order of meaningful elements," in J. H. Greenberg, ed., *Universals of Language,* 58–90. Cambridge, Mass.: M.I.T. Press.

Grimshaw, J. (1979). "Complement Selection and the Lexicon," *Linguistic Inquiry* 10:2, 279–326.

Jackendoff, R. (1977). *$\overline{X}$ Syntax: A Study of Phrase Structure.* Cambridge, Mass.: M.I.T. Press.

Lasnik, H. (1976). "Remarks on Coreference," *Linguistic Analysis* 2:1, 1–22.

Lasnik, H., and J. J. Kupin (1977). "A Restrictive Theory of Transformational Grammar," *Theoretical Linguistics* 4:3, 173–196.

Lasnik, H., and M. Saito (1983). "On the Nature of Proper Government." Unpublished ms., University of Connecticut and Massachusetts Institute of Technology.

Lightfoot, D. (1982). *The Language Lottery: Toward a Biology of Grammars.* Cambridge, Mass.: M.I.T. Press.

Liskov, B., A. Snyder, R. Atkinson, and C. Schaffert (1977). "Abstraction Mechanisms in CLU," *Comm. ACM* 20:8, 564–576.

Marcus, M. P. (1980). *A Theory of Syntactic Recognition for Natural Language.* Cambridge, Mass.: M.I.T. Press.

Marr, D. (1982). *Vision.* San Francisco: W. H. Freeman and Company.

Miller, G. A., and N. Chomsky (1963). "Finitary Models of Language Users," in R. D. Luce, R. R. Bush, and E. Galanter, eds., *Handbook of Mathematical Psychology,* vol. II, 419–492. New York: John Wiley and Sons, Inc.

Milne, R. W. (1983). *Resolving Lexical Ambiguity in a Deterministic Parser.* Ph.D. thesis, University of Edinburgh.

Newmeyer, F. J. (1980). *Linguistic Theory in America.* New York: Academic Press.

Reinhart, T. (1976). *The Syntactic Domain of Anaphora.* Ph.D. thesis, Department of Foreign Literatures and Linguistics, M.I.T., Cambridge, Mass.

Robinson, J. J. (1982). "DIAGRAM: A Grammar for Dialogues," *Comm. ACM* 25:1, 27–47.

Sager, N. (1981). *Natural Language Information Processing: A Computer Grammar of English and Its Applications.* Reading, Mass.: Addison-Wesley.

Seidenberg, M. S., M. K. Tanenhaus, J. M. Leiman, and M. Bienkowski (1982). "Automatic Access of the Meanings of Ambiguous Words in Context: Some Limitations of Knowledge-Based Processing," *Cognitive Psychology* 14, 489–537.

Sheiber, S. M. (1983). "Direct Parsing of ID/LP Grammars." Technical Report 291R, SRI International, Menlo Park, California. Also appears in *Linguistics and Philosophy* 7:2.

Stowell, T. (1981). *Origins of Phrase Structure.* Ph.D. thesis, Department of Linguistics and Philosophy, M.I.T., Cambridge, Mass.