



University of Pennsylvania
ScholarlyCommons

Technical Reports (CIS)

Department of Computer & Information Science

May 1990

Computation and Linguistic Theory: A Government Binding Theory Parser Using Tree Adjoining Grammar (Master's Thesis)

Robert Frank
University of Pennsylvania

Follow this and additional works at: https://repository.upenn.edu/cis_reports

Recommended Citation

Robert Frank, "Computation and Linguistic Theory: A Government Binding Theory Parser Using Tree Adjoining Grammar (Master's Thesis)", . May 1990.

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-90-29.

This paper is posted at ScholarlyCommons. https://repository.upenn.edu/cis_reports/555
For more information, please contact repository@pobox.upenn.edu.

Computation and Linguistic Theory: A Government Binding Theory Parser Using Tree Adjoining Grammar (Master's Thesis)

Abstract

Government Binding (GB) theory, as a competence theory of grammar, is intended to define what a speaker's knowledge of language consists of. The theory proposes a system of innate principles and constraints which determine the class of possible languages and, once instantiated by the parameter values for a given language, the class of well-formed sentences of that language [Chomsky, 1981].

In this thesis, I address the problem of how this knowledge of language is put to use. The answer I give to this question takes the shape of an implemented computational model, a parser, which utilizes the formulation of knowledge of language as proposed in GB theory. GB as a theory of grammar poses a particular problem for instantiation within a cognitively feasible computational model. It has a rich deductive structure whose obvious direct implementation as a set of axioms in a first order theorem prover runs up against the problem of undecidability. Thus, if we accept GB theory as psychologically real, and thus as functioning causally with respect to linguistic processing, there seems to be a paradox: we need a way of putting our knowledge of language, represented in GB theory, to use in a processing theory in an efficient manner.

I will suggest a way out of this paradox. I propose to constrain the class of possible grammatical principles by requiring them to be storable over a linguistically and mathematically motivated domain, that of a tree adjoining grammar (TAG) elementary tree. The parsing process consists of the construction of such primitive structures, using a generalization of licensing relations as proposed in [Abney, 1986], and checking that the constraints are satisfied over these local domains. Since these domains are of bounded size, these constraints will be checkable in constant time and we will be guaranteed efficient, linear time, parsing. Additionally, the incrementality of the construction of the TAG elementary trees is consistent with intuitions of incremental semantic interpretation.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-90-29.

**Computation and Linguistic Theory:
A Government Binding Theory
Parser Using Tree
Adjoining Grammar**

**MS-CIS-90-29
LINC LAB 171**

Robert Frank

**Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104-6389**

May 1990

UNIVERSITY OF PENNSYLVANIA
THE MOORE SCHOOL OF ELECTRICAL ENGINEERING
SCHOOL OF ENGINEERING AND APPLIED SCIENCE

COMPUTATION AND LINGUISTIC THEORY:
A GOVERNMENT BINDING THEORY PARSER USING TREE
ADJOINING GRAMMAR

Robert Frank

Philadelphia, Pennsylvania

May 1990

A thesis presented to the Faculty of Engineering and Applied Science of the University of Pennsylvania in partial fulfillment of the requirements for the degree of Master of Science in Engineering for graduate work in Computer and Information Science.

Aravind Joshi
(Advisor)

Jean Gallier
(Graduate Group Chair)

Abstract

Government Binding (GB) theory, as a competence theory of grammar, is intended to define what a speaker's knowledge of language consists of. The theory proposes a system of innate principles and constraints which determine the class of possible languages and, once instantiated by the parameter values for a given language, the class of well-formed sentences of that language [Chomsky, 1981].

In this thesis, I address the problem of how this knowledge of language is put to use. The answer I give to this question takes the shape of an implemented computational model, a parser, which utilizes the formulation of knowledge of language as proposed in GB theory. GB as a theory of grammar poses a particular problem for instantiation within a cognitively feasible computational model. It has a rich deductive structure whose obvious direct implementation as a set of axioms in a first order theorem prover runs up against the problem of undecidability. Thus, if we accept GB theory as psychologically real, and thus as functioning causally with respect to linguistic processing, there seems to be a paradox: we need a way of putting our knowledge of language, represented in GB theory, to use in a processing theory in an efficient manner.

I will suggest a way out of this paradox. I propose to constrain the class of possible grammatical principles by requiring them to be storable over a linguistically and mathematically motivated domain, that of a tree adjoining grammar (TAG) elementary tree. The parsing process consists of the construction of such primitive structures, using a generalization of licensing relations as proposed in [Abney, 1986], and checking that the constraints are satisfied over these local domains. Since these domains are of bounded size, these constraints will be checkable in constant time and we will be guaranteed efficient, linear time, parsing. Additionally, the incrementality of the construction of the TAG elementary trees is consistent with intuitions of incremental semantic interpretation.

Acknowledgments

During the process of doing this thesis, many people have conspired to satisfy my various *gives* and *needs*. My advisor, Aravind Joshi, introduced me to TAG and provided me with a sounding board for these ideas at various stages of development. I thank him for his unerring sense of the big picture and for the seemingly endless patience required to keep me moving. Tony Kroch taught me much about the role of TAG in linguistic theory and about keeping a critical eye toward theoretical dogmatism. Mitch Marcus and Mark Steedman inspired me to think about the importance of computation in studying language.

Breck Baldwin, Naoki Fukui, Michael Niv, Yves Schabes, Ramesh Subrahmanyam, Enric Vallduví and Mike White all provided very helpful comments and discussion. I especially thank Jamie Henderson for his debate on almost everything in this thesis. This work is undoubtedly much improved as a result. Participants in CLiFF and in Naoki Fukui's spring '90 syntax seminar allowed me to present this work and gave much helpful feedback.

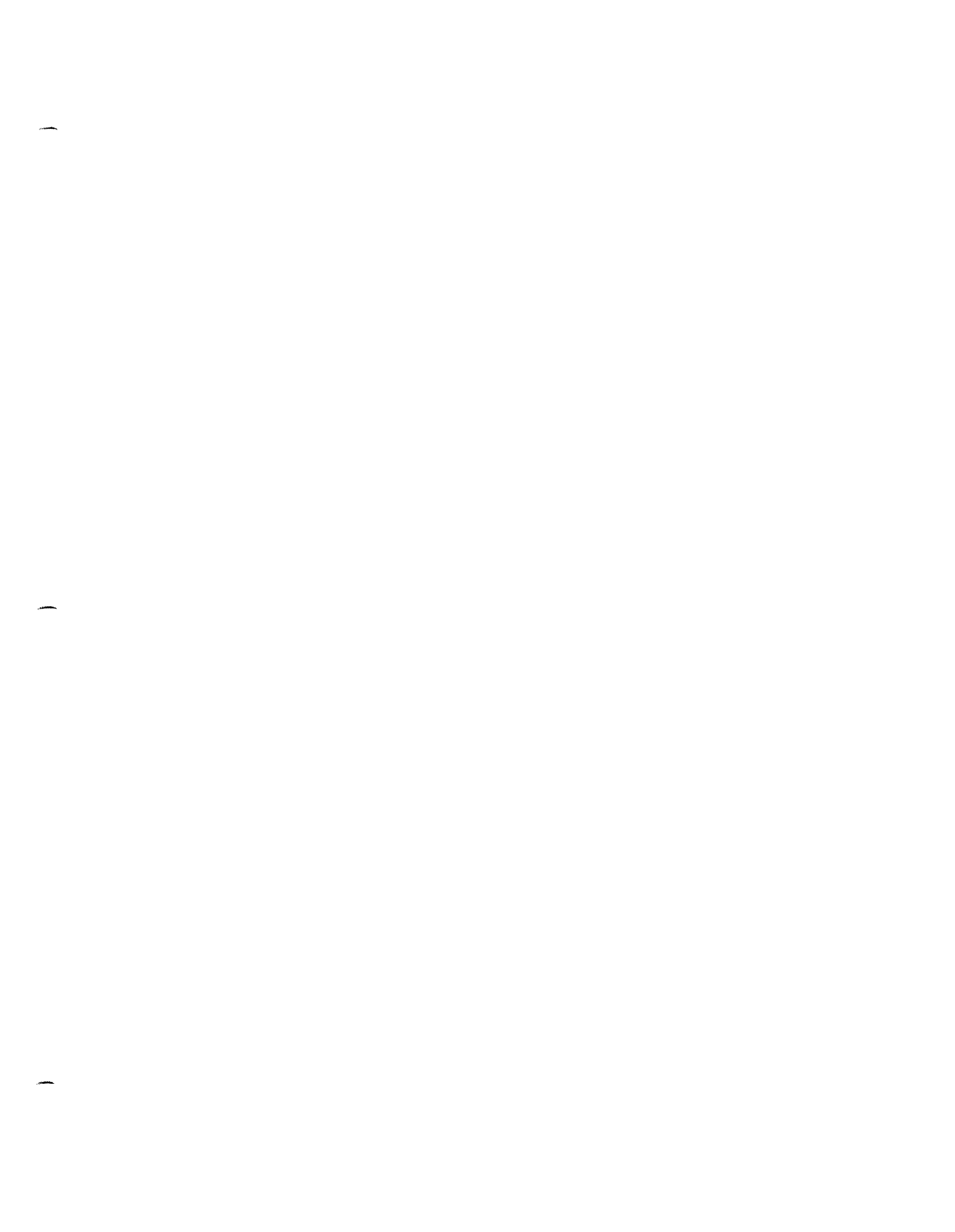
Finally, I must thank the people who kept me sane (or did they?) through this whole process. People who are especially to blame are my roommates, Beth Hudson and Kitty Meyer, the gang at 3401, and the devotees of the GSB. My family provided an ample supply of love, support and guilt, and to them is this thesis dedicated.

This work was partially funded by ARO Grants DAAL03-89-C0031 PRI and DAAG29-84-K-0061 and DARPA grant N00014-85-K-0018. The author was supported by a Unisys doctoral fellowship.

Contents

Abstract	ii
Acknowledgments	iii
1 Introduction	1
1.1 Implementations of Linguistic Theories: How and Why?	2
1.2 Knowledge of Language and Language Use	5
2 Preliminaries	11
2.1 An Introduction to Government Binding Theory	11
2.1.1 X-bar Theory	14
2.1.2 Theta Theory	16
2.1.3 Case Theory	19
2.1.4 Movement Theory	20
2.1.5 Government Theory	22
2.1.6 Extended Projection Principle	23
2.1.7 Binding Theory	23
2.2 Tree Adjoining Grammar	25
2.2.1 Linguistic Applications of TAG	27
2.2.2 Elementary Trees as Predications	35
3 The Parsing Model and Its Linguistic Implications	40
3.1 Abney's Licensing	41
3.2 Licensing revisited: Generalized Licensing	44
3.3 Using Generalized Licensing for Parsing	49
3.4 Encoding of Grammar through Generalized Licensing	51

3.5	Limiting the Domain using TAG	55
3.6	Linguistic Adequacy and Implications of the Parsing Mechanism	61
3.6.1	Raising	61
3.6.2	Control	68
3.6.3	Exceptional Case Marking and Anaphor Binding	70
3.6.4	Modification	76
3.6.5	Modifier Extraposition and Heavy NP shift	80
3.6.6	Genitives	82
3.7	Remaining problems	83
3.7.1	Incompleteness	84
3.7.2	Wh-movement and Phonologically Empty Heads	85
3.7.3	Head movement	86
4	Conclusion	87
	Bibliography	93



Chapter 1

Introduction

Noam Chomsky has proposed the following three questions as central to the study of the human language faculty [Chomsky, 1986b]:

1. What constitutes knowledge of language?
2. How is knowledge of language acquired?
3. How is knowledge of language put to use?

The research program carried out within generative grammar by Chomsky and others, though, has largely addressed only the first of these questions. Most recently, Chomsky's answer to this first question comes in the form of Government and Binding (GB) Theory: a system of purportedly biologically innate principles and modules which determine the class of possible languages [Chomsky, 1981]. These principles are parameterized by a finite set of variables which can take a small number of possible values, so the theory goes, and thus the possible human languages are those which are instantiations of the statements of universal grammar by particular parameter settings. The speaker's knowledge of the syntax of a language, then, is this set of principles and the relevant parameter values.

The second question, concerning acquisition, has received at least some treatment in that this characterization for the knowledge of language determines what a speaker must learn in order to acquire the language: parameter values. The problem of determining the values of these abstract parameters from the data given to the child, though, remains open.

In this thesis, I address the third question, how our knowledge of language is put to use. The answer I give to this question takes the shape of an implemented computational

model, a parser, which utilizes the formulation of knowledge of language as proposed in GB theory. GB as a theory of grammar poses a particular problem for instantiation within a cognitively feasible computational model. It has a rich deductive structure whose obvious direct implementation as a set of axioms in a first order theorem prover runs up against the problem of undecidability. Thus, if we accept GB theory as psychologically real, and thus as functioning causally with respect to linguistic processing, there seems to be a paradox: we need a way of putting our knowledge of language, represented in GB theory, to use in a processing theory in an efficient manner.

I will suggest a way out of this paradox. I propose to constrain the class of possible grammatical principles by requiring them to be storable over a linguistically and mathematically motivated domain, that of a tree adjoining grammar (TAG) elementary tree. The parsing process consists of the construction of such primitive structures, using a generalization of licensing relations as proposed in [Abney, 1986], and checking that the constraints are satisfied over these local domains. Since these domains are of bounded size, these constraints will be checkable in constant time and we will be guaranteed efficient, linear time, parsing. Additionally, the incrementality of the construction of the TAG elementary trees is consistent with intuitions of incremental semantic interpretation.

The remainder of this chapter discusses the slippery problem of what it means to implement a linguistic theory. I address the nature of the connection between knowledge of language and the use of that knowledge and I also consider the ways in which the study of each of these might influence our conceptions of the other. In chapter 2, I present the building blocks which form necessary preliminaries for the current effort. Specifically, I describe GB theory and the TAG formalism. I encourage readers who are familiar with these to nonetheless read this chapter, as it presents a fairly personal view of both of these frameworks. Chapter 3 contains the real substance of this work. It presents the parsing model and describes its operation on a variety of linguistic phenomena. Finally, in Chapter 4, I describe some problems for the current approach and suggest some directions for future investigation.

1.1 Implementations of Linguistic Theories: How and Why?

There has recently been interest in building parsers embodying GB theory. Usually, the author presents a parsing model and then proceeds to argue how it represents the principles

of the theory in some more or less well defined sense. Given that all of these parsers are built to satisfy a common goal, the creation of “The true GB parser”, the degree of variation among these parsers is rather surprising.

I would like to propose that this variation is not surprising at all. Each of these efforts has been motivated by different objectives and it is these differences that lead to differences in models. What are the possible motivations one might have for such an endeavor? Some possibilities are:¹

Theory Testing Developing an implementation of a theory forces unanswered questions and holes in a theory to be addressed. Additionally, a workbench could be provided for a linguist to more easily investigate the implications of local changes to global coverage of the theory.

Perspicuity of Grammatical Representation: Practical Applications By utilizing GB theory, the grammar writer could be freed from specifying a large amount of redundant information as might be required in, say, a context free grammar. The theory provides for a perspicuous representation of the grammar and abstracts out influences of linguistic universals. Thus, the creation of a multi-lingual natural language interface would be facilitated.

Psychological Modeling of Language Use This parser might provide a psychologically plausible mechanism with which to use the principles of grammar. Such a program would constitute an answer to Chomsky’s third question of how knowledge of language is put to use.

Each of these presents different demands upon the processing model. If theory testing is a primary concern, then the directness of the relationship between the grammatical principles and representations and their implementational counterparts is of utmost importance. Only if such a direct mapping is present will a linguist be able to *directly* utilize the performance of this computational engine to help assess the status of the theory. Otherwise, shortcomings in the operation of the parser can be attributed to lack of faithfulness of the implementation

¹I do not intend to suggest that any of the parsing models which I discuss focuses entirely on one of these goals. For example, nearly all of the models are concerned with providing a rigorous formulation of the theory. In fact, any such implementation will be forced to do this to some extent. I want only to show how differences in concerns leads to different design decisions. Thus, the answer to the question “What constitutes a real GB parser?” makes no sense without relativization to the motivation for such a machine’s creation.

and not to the theory itself. Additionally, with a direct relationship between processor and grammar, any changes to the theory will be most easily incorporated in such a framework. The parsers of [Johnson, 1988] and [Stabler, 1990] fit into this category. In each case, they have axiomatized some version of GB theory into statements in a logical language, Horn clauses for Johnson, full first order logic for Stabler.² These logical formulae are given to a theorem prover, which is then asked whether a certain sentence can be given well formed representations at the relevant syntactic levels. No one principle or constraint is regarded as more primitive than another. They all function together to determine well formedness. Thus, the deductive structure of the theory is exactly mimicked by the computations of the parser. Additionally, changes in the theory simply require changes in the axiomatization rather than any change in the computational mechanism.

Parsers which are concerned with issues of perspicuity of representation include those of [Kashket, 1987] and [Dorr, 1987]. Kashket's goal is the creation of a parser for Warlpiri, a free word order language. He creates a set of representations which facilitate the statement of generalizations about such a non-configurational language.³ Thus, he does not have to exhaustively list all word order possibilities and he is able to easily enforce requirements such as the auxiliary second requirement. The word order possibilities result from the interaction of the various constraints upon the representation. His parser is designed to efficiently construct and manipulate such representations. Dorr has created a machine translation system in which each of the different grammars for the translated languages is represented by its lexical entries and its set of parameter values. This compact representation of grammar enables her to easily add new languages to the translation system in so far as the set of parameters is adequate. In both Kashket's and Dorr's systems, the representation of the principles of the grammar are not so transparently related to the parsing mechanism as in the logic based approaches. Changes in the underlying linguistic theory might potentially necessitate large changes in the parser. Additionally, the equality of all constraints in filtering out ungrammaticality is not preserved here. In both systems, structure is first constructed based upon information contained in some subset of the principles. Then, these structures are filtered out using the other principles as well formedness checks. This does not exactly conform to the deductive structure of GB theory in which all the principles

²It is not obvious that either of these logical languages are sufficient in expressive power to state the principles of current GB theory. The Least Effort Principle of [Chomsky, 1989] appears to require the use of higher order logic.

³And generalizations about languages in general, he proposes.

together conspire to determine the fate of a sentence's grammaticality. I should point out that these variations from direct implementation are not inherently bad. Rather, they simply suggest a different set of concerns in creating an implementation of GB theory from those important to theory testing and implementation transparency.

The final motivation which I mentioned for building a GB parser is to provide a model for language use. Here, there has been a real mixed bag of proposals. The reason for the lack of agreement on how this ought to be done arises, I think, from the lack of consensus on the nature of the relationship between knowledge of language and the use of that knowledge. In the hope of clarifying this a bit, I will discuss this issue in the next section.

1.2 Knowledge of Language and Language Use

A speaker's knowledge of language is essentially a system of rules which determine the properties of the language. This competence grammar is to be contrasted with the performance grammar, the system for the use of that knowledge, which specifies how language is processed. The performance grammar must act in accordance with this rule system. It must obey the rules posited by the competence grammar. The main question I want to address here is what is the connection between the two.

Let's consider some possibilities. Take the behavior which is evidenced by the pencil sitting on my desk. If I push it off of the edge, it falls to the ground. Moreover, it falls with a particular velocity, acceleration and so on. In so doing, it is acting in accordance with the laws (rules) of physics. Thus, we might view the pencil as a mechanism for "physics use". This is clearly not the proper sort of connection relevant to knowledge and use of language. The pencil is acting in a manner consistent with the rules of physics, but these rules are not being put to use by the pencil in determining its action. In no sense does the pencil internally encode the rules of physics. It is performing rule governed behavior as opposed to rule following behavior. Knowledge of language, on the other hand, is assumed to be a result of the human biological and psychological endowment and the causation on the behavior of the linguistic system must come from within.

Next, look at the case of an electronic calculator.⁴ By punching a sequence of key strokes, I can see that it is acting in accordance with the rule system of Peano's axiomatization of arithmetic, assuming I take a proper interpretation of the labels on the keys and the numbers

⁴I thank Mark Johnson for pointing out this example to me.

in the display. In this case, it is the internal structure of the calculator which causes its behavior. Here, it is less clear whether or not the calculator is performing rule governed or rule following behavior. The calculator has been constructed in such a manner so that it has no choice but to act in accordance with Peano's axioms. Is it really Peano's axioms that the calculator is following though? If we were to have an extensionally equivalent theory of arithmetic T' , or at least a theory extensionally equivalent on the subset of arithmetic which the calculator can compute, would we be any less justified in saying that the calculator was following T' as opposed to Peano's rules? It would appear not.

Then, however, the calculator is no longer a valid analogy to our linguistic system. Chomsky has argued that we should adopt a realist stance with respect to linguistic competence.

Evidently, we will try to choose among "extensionally equivalent" theories of the state attained ... that coincide on "all the evidence" but differ in depth, insightfulness, redundancy and other characteristics. This is just standard scientific practice. There is no general reason to doubt that these efforts deal with questions of fact; and apart from empirical uncertainties, there is no reason to hesitate to regard their conclusions as (tentatively) true of the language faculty. ([Chomsky, 1986b], p. 250)

By assumption, then, linguistic competence will have a place in our cognitive ontology in the same way that a processing mechanism is an object in our cognitive endowment. Hence, extensionally equivalent theories of linguistic competence are not to be considered equivalent. The theory of acquisition which is assumed to go along with GB theory makes this quite clear: the learning process consists of determining exactly the values of the grammatically specified collection of parameters and not some functional equivalent.

The models of our linguistic theories (using these words in their technical sense) must capture the *intensional* properties of these theories. We must maintain some mapping between the model and the theory such that the theory is represented and recoverable from the model. One way of accomplishing this is to associate with each theory a canonical model which is different for all theories. This notion of canonical model has a precise characterization in the literature of programming language theory. The sort of mapping that might be employed between the linguistic theory and the model, though, might be much more indirect. In fact, it might turn out that the calculator which I considered above is, in this strong intensional sense, a model of Peano's arithmetic and not some

extensionally equivalent theory as a result of some obscure mapping between theories and models. Nonetheless, the point of the example remains. The model for our linguistic theory must remain in some sense intensional so as to distinguish extensionally equivalent theories.

Finally, let us consider the case of a law abiding resident of the United States. On each April the fifteenth, the IRS will have in their files an income tax return from this good person. This person is acting in accordance with the rules of the US tax laws. Moreover, this person most certainly has an explicit representation of this law within their mental state which is causing them to act in the manner which they do. Thus, this is true rule following behavior. However, this conscious following of rules is a bit too strong for our language case. Certainly, it is false that people consciously act in accordance with the principles of grammar. Were this the case, the study of linguistic competence would be far easier. In addition, we do not necessarily require that there be such a transparent causal link between the knowledge of the rule system and the use of that knowledge.

What are we left with now? We want explicit representation of the linguistic competence within our cognitive endowment. Yet, we do not require a transparent link between the use and knowledge of language. We do, after all, want to attribute certain performance effects to the slight mismatch between the competence grammar and mechanism for use. The relationship between these two, however, cannot be arbitrarily distant if Chomsky's research methodology is to be maintained:

Two theories ... might yield the same judgments of grammaticality or form-meaning correspondence (or any other subset of relevant facts), but yet differ in that one is a better theory and/or accords better with other evidence ... There are innumerable ways in which this might happen ... we might find relevant evidence from the brain sciences to select between G and G'. In short, we are trying to discover the truth about the language faculty, opportunistically using any kind of evidence we can find ... (ibid, pp. 249-50)

He is arguing that we can adduce various sorts of evidence to assess theories of linguistic competence. Certainly, evidence of a processing nature might find its way into such discussions. However, all of this will be impossible if the relation between grammar and processor is sufficiently indirect. A lack of causal role between the nature of the competence grammar and the processing mechanism would render useless any attempts to discover facts about competence from external data (i.e. data other than grammaticality judgments and form-meaning correspondence on which the theory has been constructed). Otherwise, I

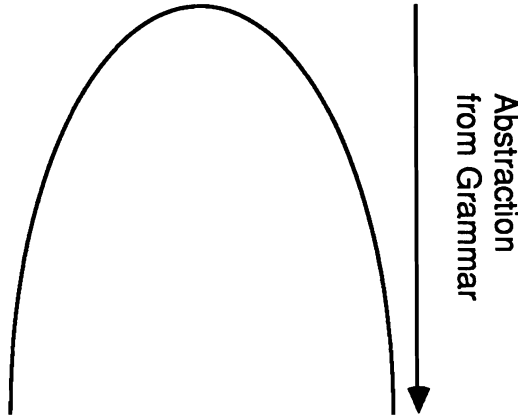


Figure 1: Space of models of linguistic processing

cannot imagine what sort of “evidence from the brain sciences” might prove informative short of the possibility of the reading off sentences in the language of thought from our mental state, an unlikely possibility. Thus, we need to investigate theories of language use which are substantively influenced by the shape of linguistic theory. If such a processing theory is correct, it will allow us to explore the predictions made by the competence theory on linguistic behavior.

What, then, does it mean for a theory of language use to be substantively influenced by the shape of the linguistic theory? This is not intended to be a binary-valued notion. Instead, I propose that processing models can be (partially) ordered with respect to this property. Figure 1 illustrates this. At the top point of this hierarchy, we have transparent implementations of the competence theories like the logic based approaches mentioned above. At a lower point of this hierarchy, would be implementations of competence grammar through compilation of the principles into, say, a context free grammar.

Of course, this influence from competence is not the only requirement we have in the construction of a theory of language use. This theory must accord with real language processing data. Certain basic requirements should hold of the model in order to satisfy what we might call psychological plausibility. Thus, since human language processing is an effortless process, we should expect that this should take place efficiently. This consideration alone rules out the logic approaches as models of use. In addition, we should expect the mechanism to operate in a fairly incremental fashion. This models the intuition that as

we hear a sentence, we build up a representation of the meaning without waiting until it is completed. Another argument for incremental processing is that we can perceive ungrammaticality almost immediately after the ill formedness occurs and thus we should allow our processing mechanism to balk on ungrammatical input before the entire sentence is considered. Once we have a processing model which satisfies these basic psychological plausibility requirements, we should evaluate it with respect to actual experimental data.

The methodology I suggest that we adopt in searching the space of possibilities is to start from the top of the hierarchy and move downward, at each point considering whether the model satisfies the psychological plausibility requirements and so on. If we succeed in this search, we will be guaranteed to have as direct an instantiation of the competence theory within the performance model that is possible subject to the constraints of processing.

Once we reach the point where we have such a model, we can consider the impact of this computational model on the linguistic theory. At least two kinds of influence are possible in this direction. [Marcus, 1980] points out that if the mechanisms which enforce the constraints of our grammar are independently motivated from computational considerations, then the degree of explanatory power of the constraints is increased. These constraints will then have two independent motivations - one from processing considerations and the other from strictly linguistic considerations - and this will constitute more evidence for the existence of such an abstract principle. In Marcus' case, he showed that the assumption of a strong computational constraint in parsing, determinism, at least in part forced him to adopt a mechanism which automatically enforced several of the constraints imposed by the competence grammar: subjacency, the specified subject constraint, and the complex NP constraint.

Another sort of impact which a processing theory could have on the competence theory is the determination of what sorts of constraints might be efficiently enforceable. Thus, a meta-theoretical constraint might be imposed on work in competence grammar: provided that constraints that are proposed as part of the competence grammar remain of a particular form, there is a model for the use of this competence that meets certain processing/psychological requirements. Suppose we have a model which achieves efficiency by exploiting a regularity in the types of grammatical principles which have been proposed thus far. This regularity, then, has independent motivations from processing and linguistic considerations. It is motivated by processing in so far as it allows for efficiency in our mechanism. The linguistic motivation comes from the observation that such a class of principles

has proven adequate to characterize grammatical knowledge.

In this thesis, I will present a processing model which has been constructed within this methodological framework. It tries to represent the principles of competence grammar in as direct a manner as is possible while maintaining efficiency. Moreover, it attempts to forge a connection with linguistic theory in the second manner discussed. That is, I exploit a locality property of the grammatical constraints as enforced by the TAG formalism to guarantee efficient processing and to maintain psychological plausibility.

Chapter 2

Preliminaries

As I said in the previous chapter, this thesis is an attempt to understand the connection between the grammar which a speaker has of a language and the mechanism which the speaker employs in utilizing this grammar. The theory of grammar with which we will concern ourselves is the Government Binding theory of [Chomsky, 1981]. In the first part of this chapter, I will present a brief introduction to this theory.

The second part of this chapter will be devoted to the Tree Adjoining Grammar formalism. The use of this formalism as the language for the specification of our theory of grammar will be crucial to the maintenance of computational efficiency and psychological plausibility.

2.1 An Introduction to Government Binding Theory

Government Binding (GB) Theory is a competence theory of grammar which attempts to categorize the class of possible natural languages. It does this by positing a set of linguistic principles and constraints which are taken to hold universally across human languages. This set of principles, the so-called Universal Grammar (UG), is augmented by a set of language specific parameters and a lexicon, the Particular Grammar (PG) of a given language. UG and PG together constitute the knowledge which a speaker is said to have when he knows a language. UG is assumed to be part of a child's innate cognitive endowment and thus syntax acquisition consists of fixing the values of a small finite set of parameters.

Each of the principles of GB theory is rather simple. They make statements about

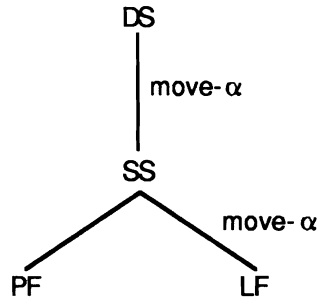


Figure 2: GB model of grammar

requirements of syntactic representations using the common vocabulary of government, c-command, and binding among others. The principles are rather heterogeneous and function independently to constrain grammaticality. This simplicity is rather superficial, though. GB derives its explanatory power from the complex interaction of the independent constraints.

The model of grammar which is assumed in the theory consists of 4 levels of representation: D-Structure (DS), S-Structure (SS), Logical Form (LF) and Phonological Form (PF). DS is taken to be a pure representation of predicate-argument structure and serves as the interface between syntactic representation and the lexicon. LF is the level which is utilized in semantic interpretation. Issues such as scoping and quantification are resolved here. PF provides the connection between syntax and phonology. SS represents the surface syntactic form and is the locus of interaction among the levels. The structure of these levels is shown in figure 2. D-structure and S-structure are related by a generalized movement operation called Move- α which allows the movement of any constituent element to any place in the structure. Between S-structure and LF, there is another instance of the Move- α relation. Finally, SS and PF are related by various deletion and movement rules.

Any sentence will have representations at all of these levels. It is judged grammatical by the theory if and only if each of these levels are well formed and are related to one another in the appropriate manner. The well-formedness requirements on each of these levels are different since the various modules and principles of grammar apply in different ways to the different representations. Some principles will apply only at a single level, while others will apply in slightly mutated forms. The levels of representation are related in the appropriate way if they are produced by the operations mentioned above which transform one level to

another, move- α between DS and SS for example. Additionally, the levels are constrained to adhere to the *Projection Principle*:

- (1) Representations at each syntactic level are projected from the lexicon, in that they observe the subcategorization properties of lexical items.

Thus, we may not arbitrarily eliminate or alter elements of the representations.

The guiding force in the creation of GB theory has largely been Occam's razor: the theory which requires the least stipulation and provides the most explanation is best. As a result of this, the methodology taken in the field has consisted of two, not necessarily independent, activities. One has been the consideration of data from a variety of languages and verifying that the theory as it stands accounts for this data by amending the theory in the appropriate way. The other has been the minimization of the number of independent principles which are required to account for the already known data. One principle or module is shown to be subsumable under another one or through the interaction of several others.

Clearly, such a methodological stance leads to a dynamically changing theory. Thus, it is never clear exactly what true GB theory is. In the remainder of this section, I will present one of many possible current versions of the theory. More regrettable, however, is the lack of definition as to what constitutes a valid "GB theory". There is a vague notion that the central ideas of government and simple phrase structural relations should constitute the terms in which the principles are stated. However, the class of possible principles is never clearly delineated. The set of tools that a linguist may use in constructing the theory are left to his imagination. Since linguistics is an empirical enterprise, it is not a priori obvious that the construction of such a meta-theory of grammar is possible. However, in the next chapter, I show that the consideration of processing issues leads to a restriction on possible grammatical constraints and thus constitutes a method for eliminating possible principles as candidates for elements of the theory of grammar.

In the remainder of this section, I will briefly discuss some of the relevant modules of GB theory, the principles which serve to constrain the various syntactic representations. For a more detailed introduction, I refer the reader to [van Riemsdijk and Williams, 1986] or [Lasnik and Uriagereka, 1988].

2.1.1 X-bar Theory

It had long been assumed that grammar consisted of a separate base component which generates a set of primitive structures on which various transformations and constraints would act. This base was usually expressed as a collection of context free grammar productions.

$$(2) \quad PP \rightarrow P NP$$

When writing such a base, grammar writers invariably adhered to certain patterns yet there was no explicit constraint on the form these rules. For instance, the absurd rule

$$(3) \quad VP \rightarrow PP A N$$

is as valid a part of the base as the NP rule given above. X-bar theory is an attempt to capture cross-categorical generalizations about the phrase structure of a language.

What sort of generalizations are there to be captured? There are two. The first is that all phrase structure results from a process of projection. Notice that all phrasal nodes dominate a unique element of *zero-level category* (i.e. of category N, P, A or V) and of the same type. That is, a PP dominates one and only one zero-level category, a preposition. This zero-level element is called the *head of the projection*. So, any XP can be thought of as determined by a single head of like category X. All of the other nodes dominated by the XP in the tree will be other phrasal nodes with their own heads. As [Stowell, 1981] has shown, the categories of these nodes, which were explicitly given in a context free base, can be determined from other independent principles of grammar. Thus, a separate base as a component of the grammar is redundant and unnecessary.

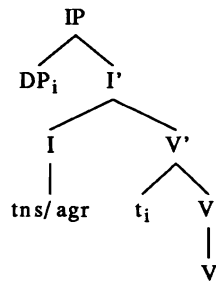
We say that this XP node is the *projection* of X. Thus, our PP production stated above is licit in that it has a unique head P which it dominates and all other dominated nodes are independent projections. In contrast, the VP production lacks a V head and also dominates two distinct zero-level categories. We can encode this idea of projection and headedness via the following schema:

$$(4) \quad \begin{array}{l} \text{The X-bar Schema (order irrelevant)} \\ X'' \rightarrow Y''^* X' \\ X' \rightarrow X Y''^* \end{array}$$

The second regularity is one which can be observed across the phrase structure of a single language. If we look at the directionality of the head with respect to its complement (the sister of X in the schema) or with respect to its specifier (the sister of X' in the schema),

we find consistency across the structures of a language. That is, a head will always precede or follow its complements and likewise for the specifier with respect to the X' projection. Thus, the schema given above may be further instantiated for a given language to include directionality. This directionality will constitute a parameter of grammar. For example, in English, the specifier appears to the left of the X' while the complements appear to the right of the head. A child learning a language now need only learn the directionality for one type of categorial projection and generalize from that single case or she can use the directionality of all projection types to help infer the directionality parameters for the language.

So far, we have assumed that the X-bar schema applies only to lexical categories N, P, V and A. What, then is the X-bar theoretic status of S or S' ? They do not seem to fit neatly into the “projection of a head” as proposed above. Recently, there has been some suggestion that the X-bar schema should be expanded to include S and S' type projections. These are taken to be projections of the *functional categories* I(nflection) and C(complementizer) respectively. Additionally, to account for the fact the functional category D(eterminer) projects to its own phrase¹, D is taken to be the head of DP (what is usually called NP) and the projection of NP is its complement. Functional categories are defined as those categories which do not possess an associated set of thematic roles. A sentence is thus a projection of I whose specifier is its subject, a DP perhaps, and its complement is a VP.



Notice that in the X-bar schema given above, there is no constraint on the number of specifier positions in a given projection - the specifier is Kleene starred and thus we may get zero or more. However, there seems to be a sharp contrast between the iterability of specifiers for functional and lexical projections.

¹In contrast to the illicit production:

NP → D N'

- (5) i. the very very old man
 ii. Mary's big red book
 iii. Susan never could have been eating cabbage.
 iv. * the the old man
 v. * yesterday's chomsky's book
 vi. * It Mary ate a bagel.
 vii. * What who did buy?

In these examples from [Fukui and Speas, 1986], we see that the lexical heads (A, N and V) allow iteration of their specifiers while the functional heads (D, I and C) do not.² Fukui and Speas therefore propose Relativized X-bar Theory in which functional and lexical categories are projected differently. The Relativized X-bar theory states that lexical categories are projected to a single bar level which may be indefinitely iterated, yielding an unbounded number of what we have been calling specifiers. Functional categories, on the other hand, are projected to two bar levels as in the original X-bar schema, but the specifier is restricted to be unique.³

2.1.2 Theta Theory

Theta theory is intended to account for the logical notion of thematic and argument structure within GB theory. The objects which theta theory is concerned with are *theta roles* and *theta grids*. Theta roles are the semantic relations which hold between a predicate and its arguments. Examples of these include such things as agent, location, experiencer, source, theme. The lexical entry of a predicate contains a theta grid which specifies the theta roles assigned by that predicate. A theta grid is a sort of subcategorization frame. It is an ordered list of theta roles assigned by the predicate along with a specification of the way in which these roles must be syntactically realized. The theta grid might, for instance, specify that a given theta role is to be realized by an element of type DP in direct object position. One of the theta roles in a theta grid is usually distinguished and called the external theta roles. The others are called internal theta roles.

²There are other ways of explaining the ungrammaticality of the iteration of specifier on IP, via the case filter, say. However, these examples strongly suggest that the generalization about iteration possibility is correct.

³Actually, this is a bit of a simplification. Functional categories only project to two bar levels when Case is assigned to the specifier position. We'll return to this in the section on Case theory.

These theta grids are utilized in the statement of the major principle of theta theory, the *Theta Criterion*:

- (6) i. Every argument must be assigned (exactly) one theta role.
- ii. Every theta role must be assigned to (exactly) one argument.

This principle essentially guarantees that the syntactic structure directly reflects certain lexical properties. At what levels does this principle hold? We can view the aforementioned Projection Principle as requiring the theta criterion holds at all syntactic levels of representation since theta role assignment.

The statement of this principle is not entirely precise. We have yet to define the notions of argument and of theta role assignment. Argument is defined both as a category inherent notion and as a structural one. On the category inherent side, a DP is always taken to be an argument. For the structurally defined portion, all projections which appear in A(argument)-positions are arguments.⁴ In order to understand the mechanism of theta assignment, we must first give the definition of government:⁵

- (7) α governs β if and only if
 - a) α m-commands β and
 - b) there is no γ , a barrier for β , such that all segments of γ dominate β and α m-commands γ .
- (8) α c-commands β if and only if
 - a) α does not dominate β and
 - b) all nodes dominating α also dominate β .
- (9) α m-commands β if and only if
 - a) α does not dominate β and
 - b) all nodes dominating the maximal projection of α also dominate β .

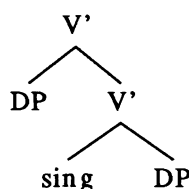
This definition of government is a formalization of the traditional grammatical notion. It is intended to represent the structural relationship between an element and those elements which are “close” to it and in some sense licensed by it. C-command and m-command are structural relations. C-command essentially roughly requires that the first node dominating

⁴This is intentionally quite vague. The notion of argument as a structural notion seems to be a *reductio*. However, we don't want to require that CP always is an argument since it can appear as a relative clause.

⁵This is essentially the definition from [Chomsky, 1986a]. Chomsky defines c-command and m-command in terms of exclusion. However, the differences between the two formulations are not relevant for the current work

one element also dominate a second. M-command loosens the requirement slightly by requiring only the first maximal projection dominating an element to dominate the other. Both of these relations are non-symmetric. Thus, by depending on these structural notions, government is not a symmetric relation. Further restrictions on the government relation will make it even less symmetric. GB theory utilizes these concepts across many of its modules and as a result any change in these definitions will percolate throughout the theory. I will refrain from defining the notion of barrier as this would take us rather far afield. See [Chomsky, 1986a] and [Fukui and Speas, 1986] for further discussion. The basic idea is that a barrier is a certain type of maximal projection which prevents the government of some element internal to it from the outside. The segments of a node are all those nodes which are from the same projection and of the same bar level.

Let us return to the assignment of theta roles. Theta role assignment always takes place under government. We can further break down theta role assignment into direct theta assignment (of internal arguments) and indirect assignment (of the external argument). Direct theta assignment requires sisterhood between the theta assigner and theta assignee. The sisterhood relation is a canonical case of the government relation. Indirect theta assignment takes place when the m-command relation must be invoked. In this case, the theta marked element may c-command the theta assigner, but still appear within the same maximal projection. In such a case, we say that the theta role is assigned through (or by) the higher projection of the predicate to which the assignee stands in the sisterhood relation. For example, in this structure:



The DP which is sister to V' , the external argument, is assigned its theta role indirectly, while the other DP, the sister of V , is directly assigned its theta role.

It is easily seen that all of this requires that the arguments of a predicate be within the projection of that predicate. In the clausal structure proposed in the previous section, this requires that the external argument, ordinarily the subject of the sentence, appear within the projection of V .

All of this discussion has assumed that theta roles are assigned to unique nodes in the tree. This is true at DS. At other levels, the theta roles are carried by theta chains which

are rooted at the original DS node.

2.1.3 Case Theory

Case theory deals with an abstract syntactic property which holds of DPs. These elements are assigned a property of Case in certain structurally defined configurations. Case is an abstraction of the familiar grammatical notion. Hence, the lack of overt case morphology in English does not preclude the possibility that a DP has been Case marked.

Case is assigned to DPs by lexical heads of type V or P or by functional heads of type I or D. Not all such heads will assign Case. The assignment of Case by a lexical head will be a lexically determined property in much the same way as theta assignment: a lexical head will have an associated case grid. Assignment of case by a functional head will be determined by features of the functional head - for instance, agreement features in I will assign nominative case.

Lexical case assignment requires government by the lexical head. In addition, Case assignment is a directional relation so the DP must be on the correct side of the assigner. This directionality is parameterized across languages - in English, it is rightward. It also been proposed by [Stowell, 1981] that an additional requirement on Case assignment is adjacency, subject to parameterization.

The assignment of Case by a functional head has quite different structural requirements. This instance of Case assignment is more structurally analogous to indirect theta assignment. Here, Case is assigned through the X' projection of the functional head. In fact, the Case is assigned to the unique specifier position of the functional projection. The directionality of this type of Case assignment is independent of the direction of lexical case assignment, but rather is coincident with the direction of specifiers in a language, leftward for English.⁶

The grammatical constraint on Case assignment is the Case filter:

- (10) All phonologically overt DPs must have case.

The phonologically overt clause allows empty categories to lack case yet still be present in a grammatical structure. This requirement must hold at SS. DP movement between DS

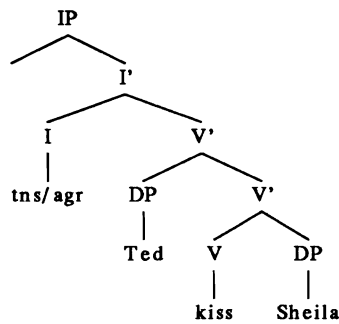
⁶Fukui and Speas propose, in fact, that such specifiers only exist when Case is assigned to them. Case is a generalization of Case, which includes wh features assigned by an appropriate Complementizer in addition to assignment of Case by lexical head and assignment of nominative or genitive case by functional head

and SS may be seen, then, as a “quest for case” by the DPs base generated (i.e. from the lexicon) in non-case-receiving positions.

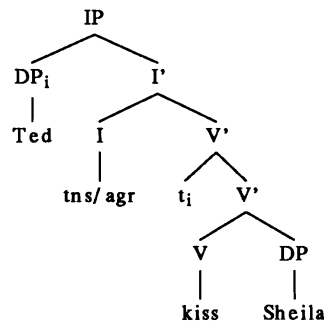
Given the discussion so far, let us consider the structure of a simple sentence:

(11) Ted kissed Sheila

At DS, the representation will be:



The external argument DP *Ted* is indirectly theta marked while the object DP *Sheila* is directly theta marked. Additionally, the object DP receives accusative case from the verb. However, the external DP does not. Thus, this representation violates the Case filter and therefore cannot also serve as the SS for this sentence. However, if the problematic DP is moved to the specifier position of I, it receives case.



The chain containing the moved DP is theta marked since its lowest element receives theta role from the verb and the theta criterion remains satisfied. Thus, we see that the interaction of Case, Theta and X-bar Theory require a fairly baroque structure for a simple clause.

2.1.4 Movement Theory

The generalized movement operation, Move- α , allows for the movement of any element to anywhere else in the structure. This movement leaves behind a coindexed trace in the

original position. As such, the operation is rather unconstrained except for its interactions with other the modules.

Various stipulations have been added which limit the applicability of movement. In particular, only maximal projections or zero level categories are accessible to the movement operation. Non-maximal projections may not be moved. Also, a constraint on the locality of the relation between the trace and moved element, *Subjacency*, has been proposed. “Classical Subjacency” states that:

- (12) No movement may cross 2 bounding nodes, where bounding nodes are DP and IP (for English)

This constraint accounts for various well-known extraction violations:

- (13) i. * Which politician_i [_{IP} did Tom see [_{DP} the book that criticized t_i]]
ii. * What_i [_{IP} is [for [_{IP} Bill to win t_i]] likely]]
iii. * Who_i [_{IP} do you know whether [_{IP} Tom despises t_i]]

In each of these cases, the movement crosses 2 bounding nodes.

So-called unbounded dependencies such as

- (14) Who_i did [_{IP} Ann hope that [_{IP} Jerry might think that [_{IP} Bill once met t_i]]]

appear to violate this constraint. However, this sort of movement is assumed to take place *successive cyclically*. That is, the moved element first moves to the spec of C position of its clause, then to the next, and so on.

- (15) Who_i did [_{IP} Ann hope [_{CP} t_i that [_{IP} Jerry might think that [_{CP} t_i [_{IP} Bill once met t_i]]]]]

In this way, each of the individual movements does not violate subjacency. Note that in the subjacency violations we saw in (13) the use of the spec of C position as an “escape hatch” is not possible. I should point out a difference between these intermediate traces and the traces at which movement begins which will be important later. The initial traces invariably appear in theta marked positions since they are present at D-structure. Thus, their presence is easily detectable and in fact required by the projection principle. However, the intermediate traces are only required so that the structure does not violate the various constraints on long distance movement, particularly subjacency. As a result, a mechanism which attempts to use such a representation for parsing will have a potentially rather

difficult problem in deciding when to posit intermediate traces and when positing further intermediate traces will not help to save the structure.

Unfortunately, this statement of the locality constraint on movement does not account for adjunct extraction violations.

(16) * Who_i did you cry when Bill saw t_i

In addition, subjacency does not distinguish two cases of extraction which differ significantly in their acceptability.

(17) i. Who_i did you see the pictures of t_i

ii. * What_i did you give the book to the girl near t_i

Thus, this formulation of the constraint is not quite correct. There has been a great deal of recent debate concerning the relationship of subjacency and barriers to government.⁷ In the next section we will discuss some of the substance of this debate, but it will remain largely outside of the scope of our discussion.

2.1.5 Government Theory

The way in which the notion of government is brought to bear on the question of locality of movement is through the *Empty Category Principle* (ECP).

(18) Every trace must be properly governed.

Proper government is defined as:

(19) α properly governs β if and only if

a) α theta-governs β or

b) α antecedent-governs β

Theta-government is government which is associated with the assignment of theta role by an X⁰ level category. Indirect theta assignment does not constitute theta-government and thus not proper government since it is not done by an X⁰ category. The conditions for antecedent government are essentially the same as those for normal government with the added proviso that the governor be coindexed with the governee as a result of an instance of move- α .

⁷See, for example, [Lasnik and Saito, 1984], [Kayne, 1981],[Huang, 1982], [Chomsky, 1986a].

The locality condition on movement can now be seen as a constraint on the relationship between the moved element and its trace. The ECP says that this element needs to be properly governed. If the movement is too far (i.e. it has crossed a barrier to government), the movement will be illicit unless the empty element is theta governed. Now, the burden of our explanation for locality rests in the definition of barrier to (antecedent) government. I will not consider here the various attempts to do this. However, note that even before we give the definition of barrier, the ECP predicts a subject-object asymmetry with respect to extraction possibilities since traces in object position are always properly governed through theta government.

2.1.6 Extended Projection Principle

The projection principle as described above guarantees that all of a predicate's arguments will be represented at all points in the syntax. Certain predicates, though, appear not to take arguments at all, the verb *rain* for instance. Nonetheless, they require the presence of a subject in syntax:

- (20) i. It rains
ii. * Rains

This insistence on predication in a clause has led to the extension of the projection principle to the *Extended Projection Principle*:

- (21) The Projection Principle holds and all clauses must have subjects.

Much debate has centered around the question of whether the “predication requirement” is reducible to other components of the theory of grammar, cf. [Rothstein, 1983], [Williams, 1980], [Heycock, 1989], [Fukui and Speas, 1986].

2.1.7 Binding Theory

Binding theory provides constraints on the coreference possibilities of nominal elements. These nominal elements are broken down into three classes: anaphors, pronouns and R-expressions. The class of anaphors is composed of reflexives, like *herself*, or reciprocals, like *each other*. Pronouns are words like *We* or *him*. R-expressions are either definite or indefinite descriptions or names. Now, binding theory states the following:

- (22) a) an anaphor is (A-)bound in its governing category
 b) a pronoun is (A-)free in its governing category
 c) an r-expression is (A-)free everywhere

Binding is defined as:

- (23) α binds β if and only if α c-commands β and α and β are co-indexed.

An element is free if it is not bound. We define governing category as:

- (24) α is the *governing category* for X if and only if α is the minimal category containing X, a governor of X, and a SUBJECT accessible to X.

Roughly, an accessible SUBJECT is the subject of either a clause or a DP (in the case of a genitive) which c-commands that element and satisfies some additional properties. I will refrain from giving a more precise definition since it is not entirely relevant to the current discussion.

The key insight in the binding theory is that the constraints should be stated in terms of disjoint reference (i.e. an element must be free) and not in terms of requirements on reference. Examples which the theory accounts for include:

- (25) i. * He_i thinks that John_i is a fool
 ii. * John_i wants him_i to leave the party
 iii. John_i wants himself_i to leave the party
 iv. John_i knows that he_i left the party
 v. * Dave_i likes Stu's painting of himself_i
 vi. Dave_i likes the painting of himself_i

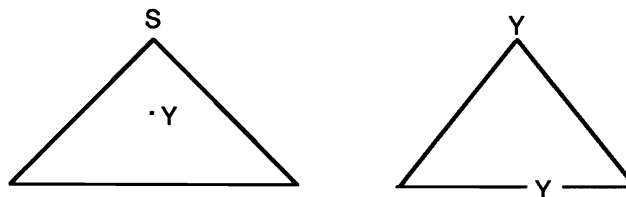
In (25)i, the r-expression *John* is bound by the subject pronoun *him*, a condition A violation. In (25)ii, the pronoun *him* is bound by *John*. The governing category for the pronoun is the entire clause since the closest accessible subject is the subject of the matrix clause. Thus, it is a condition B violation. An anaphor in the same position as the pronoun must be bound in its governing category and hence (25)iii is grammatical. In (25)iv, the agreement morphology in I functions as an accessible subject for the pronoun, so the governing category for the pronoun *he* is the lower IP. In (25)vi, the governing category for the anaphor is the entire IP since there is no accessible subject. However, in (25)v, the genitive DP *Stu* functions as an accessible subject and a condition A violation results.

2.2 Tree Adjoining Grammar

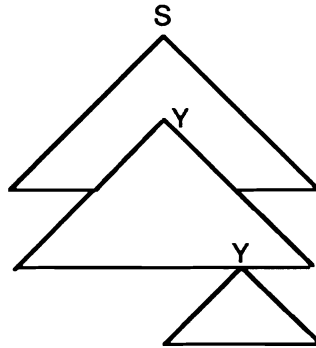
Tree Adjoining Grammar (TAG) is a constrained grammatical formalism which separates recursion phenomena from local co-occurrence restrictions [Joshi *et al.*, 1975] [Joshi, 1985]. TAG has been shown to generate a larger class of languages than the context free languages, a class known to be inadequate for natural language description [Higginbotham, 1984] [Shieber, 1984]. The increased generative capacity is, however, only slight. TAG produces the same mildly context sensitive class of languages as a number of recently developed grammatical formalisms such as head grammars, linear indexed grammars and combinatory categorial grammars [Weir *et al.*, 1986] [Vijay-Shanker, 1987] [Weir and Joshi, 1988].

TAG accomplishes the factoring of local dependencies from recursion phenomena by using a set of primitive syntactic structures over which dependencies are stated, the *elementary trees*, and a combination operation of adjunction. Adjunction takes two elementary trees and combines them to form a more complex structure. This resultant structure may in turn be combined with another structure through application of the adjunction operation.

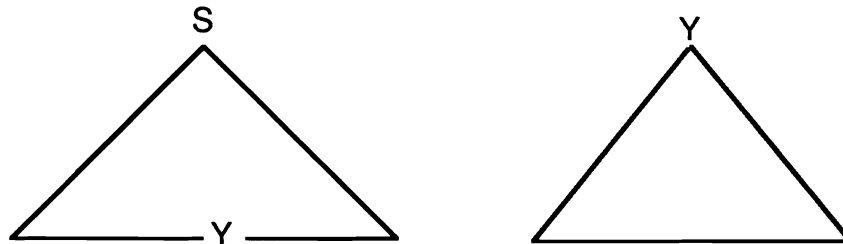
A TAG is defined as a pair of finite sets of elementary trees: $G = \langle I, A \rangle$. These two sets represent the two different type of elementary trees: initial and auxiliary. Initial trees are required to be rooted in some designated start non-terminal and to have only terminal symbols at the frontier. Auxiliary trees, on the other hand, must have a single non-terminal along their frontier of the same type as the non-terminal at the root with all remaining frontier nodes being terminals. This distinguished non-terminal frontier node is called the foot node of the trees.



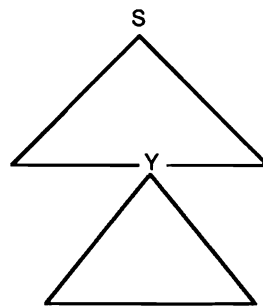
The adjunction operation allows the “insertion” of an auxiliary tree at another tree’s interior node which is of the same type as the root and foot node of the auxiliary. Thus, the previous trees may be combined through an adjunction to form the following:



This system may be extended to include an operation of substitution. We allow initial trees to be rooted in non-terminals other than the start symbol. Also, the requirement on terminal symbols at the frontiers of the elementary trees, for both initial and auxiliary trees, is loosened to allow any number non-terminals which may function only as substitution sites. In this extended system, the following trees are well formed elementary trees:



They may now be combined through substitution:



It can easily be shown that the substitution extension of the TAG formalism adds no descriptive power.

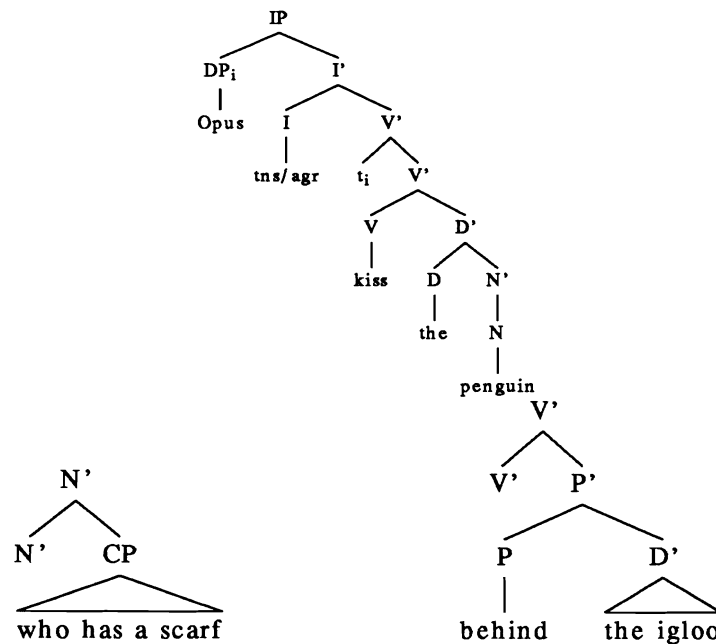
The applications of each of these operations is recorded in a derivation structure. This structure represents the trees which have been combined, through what operations they were

combined, and at what node in the trees the relevant operation was performed. Semantic interpretation is presumed to take place on the level of derivation structure. Given a denotation for the elementary trees, a compositional semantics is then stated over this derivation structure [Subrahmanyam, 1988].

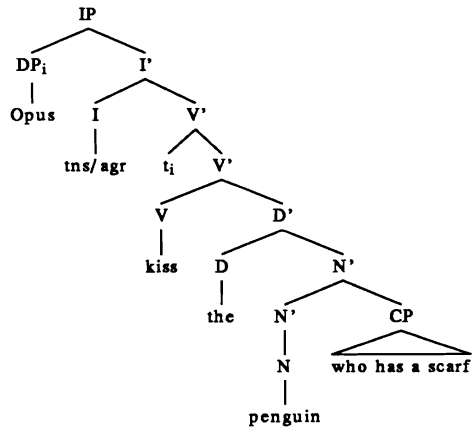
2.2.1 Linguistic Applications of TAG

TAG has proven to be rather useful for linguistic descriptions (see, for example, [Kroch and Joshi, 1985] [Kroch and Joshi, 1986] and [Kroch and Santorini, 1987]). The formalism allows the grammar writer to treat local dependencies and recursion as orthogonal. And, by providing an extended domain of locality over which constraints can be stated, the formalism allows for a natural treatment of these dependencies. Let us examine a small fragment for English to see how this is done.

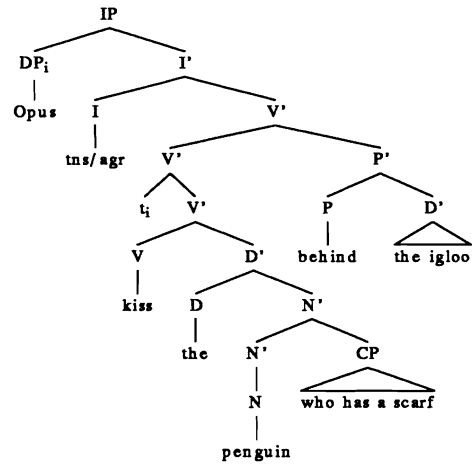
Consider the following elementary trees:



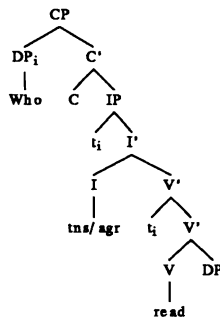
Notice that the constraint on agreement between the agr features in I and the DP subject may be expressed within the confines of the single elementary tree. Now, relative clause modification of the object DP may be accomplished through the adjunction of the DP auxiliary tree at the object DP node in the initial tree.



This more complex structure may again take part in an adjunction this time at the V' node allowing for clausal modification.



For sentences involving movement, both filler and gap appear within the same elementary tree, as was seen in the movement of the subject from its V' internal position to spec of I. For a simple question, we have the following elementary tree:

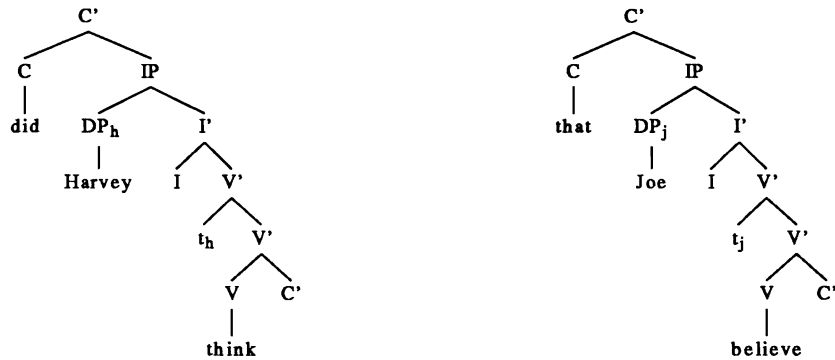


Unbounded dependencies are treated by allowing adjunction of intervening clausal structures between the trace and the moved element, thereby stretching the link between the

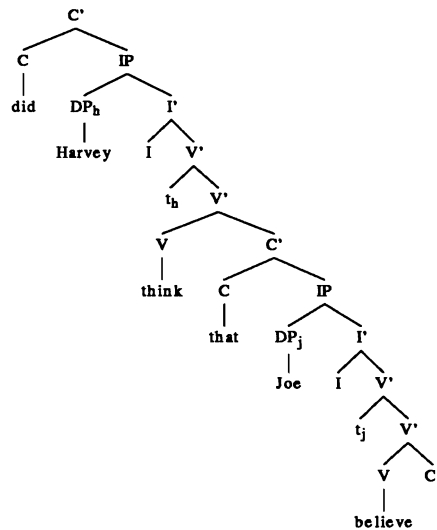
two. Now, in order to derive the sentence

(26) Who did Harvey think that Joe believed read the magazine

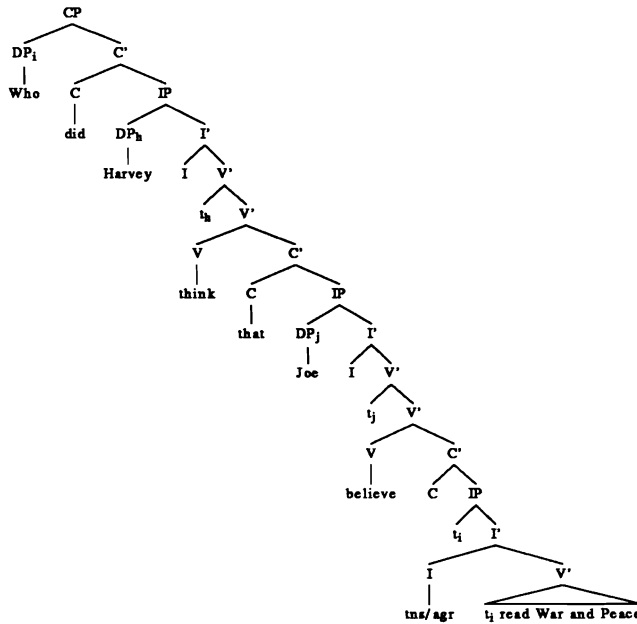
we require three elementary trees, the initial tree give above and the following two auxiliaries:



We can first adjoin one auxiliary tree into the other,



and then adjoin this whole resulting structure into the initial tree.



One might ask why we do not simply posit larger elementary trees which are of exactly this form. Thus, we would not need any adjunctions at all. Nothing we have stated about the TAG formalism itself prohibits us from doing this. However, we will want our grammar to be able to generate arbitrarily long sentences with arbitrarily long distances between the filler and gap. And, since we have required our grammar to be finite, we cannot accommodate all such cases. Thus, we can turn the question around and ask “why should the case of N clauses be handled without an adjunction, but $N+1$ with one?” There seems to be no principled answer to this question. Moreover, the attractiveness of TAG mentioned above was the factoring apart of recursion and local dependencies. This implies a view of TAG elementary trees which are strictly non-recursive structures in the case of initial trees of minimal recursive element for auxiliary trees. This corresponds intuitively to a predicate and all of its arguments. This convention has been adhered to throughout TAG linguistic work. In section 3.2.2, I will formalize this notion and provide a precise definition for the domain which constitutes an elementary tree.

Returning to the derivation for the Wh-question, let us consider other possible orders in which we might have proceeded. First, we might adjoin the lower auxiliary tree into the initial tree and then adjoined the upper auxiliary tree into this larger structure. Alternatively, we might first adjoin the higher auxiliary into the initial tree and then adjoin the lower auxiliary between the two clauses. We will not allow this final order for the derivation, though. Why? Let us assume that during a derivations, semantic interpretation takes place

incrementally and monotonically. The adjunction operation might be interpreted as a sort of function application or theta grid saturation of the predicate determining the auxiliary tree. Thus, upon performing the first step of adjunction, we are simultaneously performing theta role assignment. This theta assignment has associated semantic force and is thus subject to the monotonicity restriction. Consider the consequences of adjoining an auxiliary tree in the middle of this theta assignment configuration. By our monotonicity assumption, we must allow the theta assignment to persist despite the lack of appropriate structural configuration in the final structure. However, by allowing the first theta assignment to persist, we will have caused a theta criterion violation since the lower clause will receive another theta role from the recently adjoined auxiliary tree. Also, the argument which constitutes the newly adjoined auxiliary will not receive a theta role since the theta role from the upper clause has already assigned its theta role to the higher clause by assumption.

In past work, this sort of constraint on the ordering of TAG derivations has been enforced through the use of *adjunction constraints*. An adjunction constraint is an annotation on a node which specifies what sort of conditions on adjunction obtain there: *Obligatory adjunction* (OA) specifies that an adjunction must take place at that node; *Selective adjunction* (SA) specifies that an adjunction may take place providing that the adjoining element is an element of some specified set of auxiliary trees; *Null adjunction* (NA) disallows all adjunctions at that node. In order to accomplish the requisite ordering, we need only to place an NA constraint on the foot nodes of each of the auxiliary trees. Once an adjunction is performed, the NA constraint on the foot node is inherited by the node of interest and prevents any intervening adjunctions from occurring. A recent extension to the TAG framework, Feature-based TAG (FTAG), allows these adjunction requirements to be expressed in terms of constraints on unification of pairs of feature structures that are associated with each node [Vijay-Shanker, 1987].

Instead of viewing these constraints as primitives, I suggest that we view them as resulting from the principles of grammar. For example, as we saw above, the ordering on the adjunctions can be derived from the theta criterion. Thus, universally positing NA constraints on foot nodes of auxiliary trees is not necessary if we allow the principles of grammar to constrain the structure formed during the derivation. My suggested conception of the theory of grammar, then, consists of the TAG formalism in addition to the set of grammatical constraints described earlier. The role of TAG in the theory is to determine the domain over which such constraints will be stated. This constitutes a strong empirical claim

about the nature of the set of grammatical constraints. They must remain local in a strong, well-defined sense. This view of GB, if it can be maintained, has profound implications not only on the process of theory construction but on the properties of the mechanisms which put grammatical knowledge to use. In the next chapter, we will investigate the ramifications of such a view on a model of syntactic processing.

At first glance, this locality of constraints does not seem to be maintainable. As discussed in the first half of this chapter, certain long distance movements result in ungrammaticality. GB theory attempts to capture these ungrammaticalities through constraints like subjacency and the ECP. Since these constraints talk about relationships across large domains, one would expect that they are not expressible within the TAG formalism.

However, in two remarkable papers ([Kroch, 1986] [Kroch, 1989]), it is shown that such constraints are, in fact, expressible in TAG. Kroch's analysis covers a substantial chunk of the known data concerning such constructions. The analysis consists of a formulation of the TAG analog of the ECP as a constraint on the well formedness of elementary trees.

- (27) For X any node in an elementary tree α , initial or auxiliary, X is properly governed if and only if one of the following conditions is satisfied:
1. The maximal government domain of X is the root node of α
 2. X is coindexed with a local c-commanding antecedent in α

Maximal government domain is essentially the transitive closure of the governing category relation weakened by removing the requirement for accessible subjects. The analog of the ECP is thus:

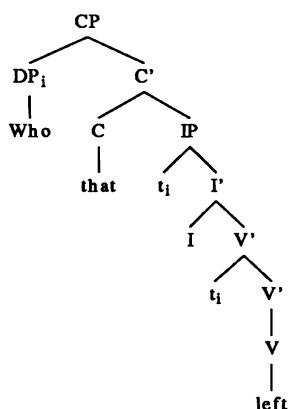
- (28) For any node in an elementary tree α , initial or auxiliary, if X is empty, then it must either be properly governed or the head of an athematic auxiliary.

A few clarifications are in order. The notion of a node being empty is equivalent to the notion of empty category for mainstream GB, with the one addition that the foot nodes of auxiliary trees are considered to be empty. Now, an athematic auxiliary tree is an auxiliary of the kind used for modification structures, the prepositional phrase and relative clause structures show above, for example. Its foot node is the daughter of the root and the recursive relation between them is formed to a Chomsky adjunction. Thus, the foot node of such auxiliaries is not theta marked. Complement auxiliaries, the other type of auxiliary tree, is identified by the fact that its foot nodes are theta marked. Now, we can see that

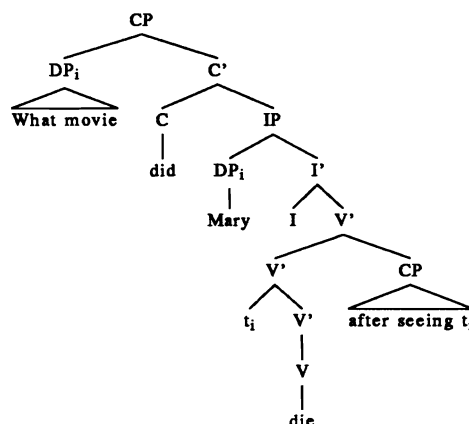
this indeed constitutes a TAG constrained statement of the ECP in that it is stated over the domain of a single elementary tree.

This formulation of the ECP is quite reminiscent of the proposals of [Huang, 1982] and [Kayne, 1981]. However, crucially different is the fact that we do not continue the search for a *g-projection* or *extraction domain* over an unbounded distance. Rather, the guarantee that an auxiliary tree is the proper sort of recursive structure guarantees us that any adjunction performed will not violate a currently well-formed link between a gap and its filler.

Now, it can be seen that the subject-object extraction asymmetry is a consequence of the subject trace not being properly governed. “That-trace effects” are captured since the initial tree required to generate them is ill formed. The subject trace in specifier of I position is not properly governed since the overt complementizer acts as a barrier to antecedent government.



Standard subjacency effects, which are taken to be independent of the ECP, are ruled out by properties of the TAG formalism itself. Extraction out of adjunctions, for example, is not possible since the formalism does not permit the construction of the required elementary tree. The filler and gap must be localized, but there is no way to define such a tree which would localize them and produce the following resultant structure:



Similarly, we can rule out extraction out of noun phrases (DPs in our terms) and sentential subjects. Lack of extraction from Wh-islands reduces to the independently necessary stipulation that any single elementary tree can contain at most one fronted Wh-element. Notice that in all of this, we do not have to make stipulations about arbitrarily sized domains, but only about elementary trees. Additionally, the stretching of the link through adjunction relieves us of having to posit intermediate traces, objects which are often theoretically troublesome and are certainly computational nightmares.

I would like to comment on the shape of the model of grammar for the version of GB expressed in TAG we have been exploring. One major difference between the TAG model and more mainstream versions lies in our lack of utilization of multiple levels of representation for the entire structure. The elementary trees are essentially parts off SS representations which may combine together in constrained ways. Move- α is assumed to have taken place already over the minimal structures. That is, we do have the DS and SS levels of representation but only at the elementary levels. These structures combine to form a level of representation similar to SS, modulo lack of intermediate traces. But, constraints of grammar cannot be stated over the entire domain and thus the structure we build by combining elementary trees is not a level of representation in the fullest sense within GB theory. The operation which combines the elementary trees to form a pseudo SS has no real analog in GB terms. It is emphatically not the move- α mapping between DS and SS.

Conspicuously absent from this model of grammar is the level of LF. LF is taken to be a translation of SS in its entirety and as such might require considerations not local to a single elementary tree and thereby prove fatal to our enterprise. Williams [1987], however, has argued that LF is unnecessary as a distinct level of representation from SS. Since SS constraints have thus far been manageable within our constrained domain, we can expect

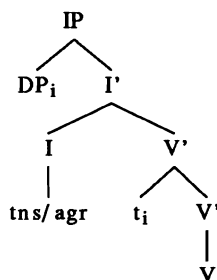
to be able to formulate his “LF” constraints in our elementary tree terms. This project remains for future work.

We have now seen that the part of GB theory concerning constraints on long distance dependencies is encodable within the TAG framework. This test case seems like a particularly good one in that it deals with a phenomenon that would *prima facie* seem to require consideration of large, unbounded domains. However, it remains an empirically testable and falsifiable claim as to whether the remainder of the theory can be so translated. Thus far, superficial investigation has suggested that it will be possible.⁸ I believe that it is advantageous for linguists to pursue this course of attempting to express the principles of grammar in a constrained grammatical framework. Such a formalism provides an explicit meta-theory to limit the class of possible principles which a linguist might propose. Currently, what constitutes a good or natural principle seems to be determined by little more than the aesthetic of people in the field. Moreover, as we will see in the next chapter, if the constraints of grammar remain in this framework, we can provide a model which can efficiently process syntactic structure utilizing these constraints.

2.2.2 Elementary Trees as Predications

In the previous section, I mentioned that the intuition behind the domain constituting an elementary tree is that of a predicate plus all of its arguments. Recent work under the rubric *Lexicalized TAGs* [Schabes *et al.*, 1988] has regarded this notion as central, but has not provided a particular formulation for this intuitive notion. In this section, I present a formalization of this ‘predicate with its arguments’ idea.

Recall the structure which we assigned to a clause, assuming the view of phrase structure presented in [Fukui and Speas, 1986]:



This structure can be seen as functionally divided into two pieces. In the projection of V, we have an unpolluted representation of the argument structure of the verb. The projections of

⁸With the one exception of Binding theory principle C: That r-expressions be free everywhere.

all lexical categories, in fact, contain structural positions for all of their arguments and these positions will be attached at uniform bar level. In a sense, then, all arguments are of equal stature in this context. This is analogous to the equality of stature among the arguments to any n-ary relation. The subject is accorded no special status here other than its being the most externally located element - a property resulting from the way in which theta roles happen to be discharged. I propose, then, to interpret this projection as a saturated logical function. Let us call this a *function-argument structure*.

Now in clausal structure, this function-argument structure is embedded within the projection of I. In contrast to the uniformity of lexical category projection, the projection of functional categories is taken to be asymmetric. This is a case of the standard X-bar schema which sharply contrasts complements from specifier. The interpretation I propose for this structure is that the I' is a predicate which is predicated over the subject that is its specifier. It is this configuration that I claim satisfies the "extended" portion of the extended projection principle. Let us call the I' dominated subtree an *unsaturated predicate* and the entire IP structure a *saturated predication*. The role of the functional element, then, is to turn a function-argument structure into an unsaturated predication awaiting its subject.⁹ The uniqueness of the specifier position now follows from semantical considerations: predicates by their very nature may take only one subject.

This analysis makes the dual role of the DP specifier of I quite clear. In its DS position, it serves as the argument of the lexical predicate, while at SS, it functions as the subject of the sentential predication.

Let us extend this functional analysis of clausal phrase structure to include all projection types. A function-argument structure will in general be the projection of single lexical head with all of its argument positions. A predication is composed of a lexical projection (a function-argument structure) and optionally its associated functional projections. In the case of V, these associated projections include I and C. For N, the only associated functional projection is D. For A and P, however, there are no such associated functional projections and the lexical projections of these categories will serve as predications in their own right, in small clauses for example. Multiple functional projections within a single predication, as in the case of V with the projections of I and C, seems slightly incompatible with the

⁹In previous work [Frank, 1989] [Frank, 1988], I have addressed the technical problem of binding the "placeholding" empty category within the function argument structure, typically the trace of the external argument, by the subject of the predication. This has been done both in a combinatorial categorial grammar setting [Steedman, 1987] as well as within Higginbotham's theory of semantics [Higginbotham, 1985].

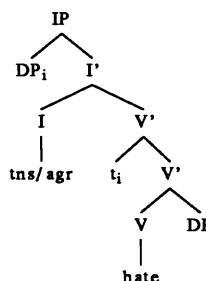
semantic characterization I am giving. However, if the semantic type of C specifies that it takes a saturated predication into an unsaturated one, then any predication involving C will necessarily contain its associated IP predication.

There is one final way in which a predication may be created. If we have an unsaturated predication P which is maximal in the sense that there is no functional category which may take P as its complement and turn it into another predication P' , the P may predicate of an entity by Chomsky adjoining to that entity's maximal projection.¹⁰

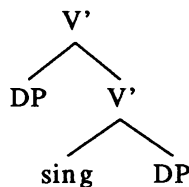
Now, we can precisely state what constitutes an elementary tree.

(29) An elementary tree consists of exactly one predication

Note immediately that this definition of elementary tree forces us to adopt the extension of the TAG framework that includes substitution. All arguments of a lexical head will constitute independent predications. In order to incorporate them into a structure which is not recursive of the correct type, they must be substituted along the frontier. Thus, one of the elementary trees for a simple clause whose lexical head is a transitive verb could be:



Predications consisting of simply the projection of a lexical head correspond to small clause elementary trees as seen in the complements of causative and perception verbs

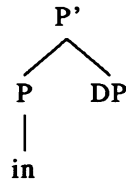


to derive things like

(30) Phillippe heard [_{V'} Marie sing an Aria]

or as substituted trees for, say prepositional complements

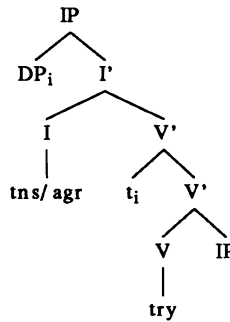
¹⁰That is to say, it must be of category D, C, A or P. Thus, we predict that projections of V, N and I will never be adjuncts. I am not sure of the reasons for the constraint that there cannot be any further associated function “jacket”, but it appears to capture the facts.



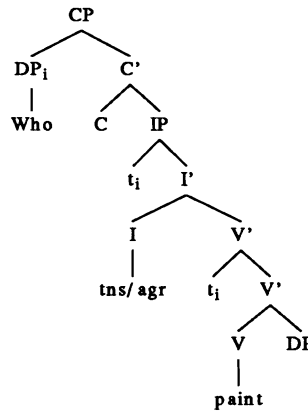
in order to yield

(31) Alex put the needle [p' in the haystack]

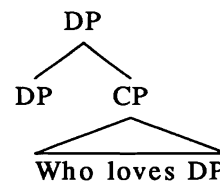
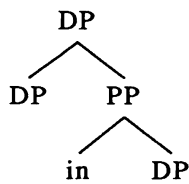
In the case of lexical heads which select for sentential complements, the empty node on the frontier representing the position of the complement determines the foot node of the auxiliary tree determined by this prediction thereby forming a complement auxiliary.



Initial trees for Wh-movement are simple predications rooted in CP:



The final class of predications, those which are Chomsky adjoined to some appropriate maximal projection, form the athematic auxiliaries used for modification structures.



This specification of the class of elementary trees provides us with a precise notion of

domain of locality in TAG. All of the analyses proposed in the papers on the linguistic applications utilize only trees of the sort I have allowed. Moreover, with this formulation, the claim that grammatical constraints can be expressed within the TAG framework becomes something quite substantive and falsifiable. Before this, the domain could always be amended so as to allow for any slight extensions that might be “necessary”. Now, the class of possible elementary trees is clearly defined. In the next chapter, I will crucially utilize this precise definition of TAG elementary tree to constrain the workings of the parsing model.

Chapter 3

The Parsing Model and Its Linguistic Implications

This chapter describes the operations of the parser. As mentioned in chapter 1, this parser aims to provide a psychologically plausible mechanism for putting the knowledge a speaker has of a language, the grammar, to use. The representation of knowledge of language which I am assuming is that specified by Government Binding Theory. Recall that the methodology I proposed in Chapter 1 forces us to maintain as direct a link as is possible between the representation of the knowledge of language and the operation of the parser while still maintaining certain desirable computational properties.

The difficulties in adopting the most natural and direct link are fairly apparent. As described in Chapter 2, GB theory posits a set of principles and parameters which serve as well formedness constraints on constituent structures and relationships between lexical items and their associated constituency. A direct instantiation of these might be encoding them as a set of first-order logical formulae. The parsing problem is then equivalent to determining the satisfiability of a predicate which assigns structure to the input string subject to the well formedness conditions of the grammar. This is the view taken by [Johnson, 1988] and [Stabler, 1990]. Of course, such a parser will inherit all of the computationally atrocious properties of first order theorem provers, such as undecidability. It is clear, then, that such an approach does not meet our requirements for psychological plausibility.

An interesting step was made by Abney [1986] in the use of *licensing relations* as the foundation for a GB parser. These relations are able to encode many of the abstract constraints in GB theory. Moreover, using such relations allows Abney to construct a parsing

mechanism with fairly computationally efficient and thus psychologically plausible properties. I believe, though, that Abney's licensing relations abstract away from the grammar a bit too much. This is methodologically troubling for reasons mentioned in chapter 1. Additionally, certain constraints of grammar are not naturally expressible within this system and the enforcing of these constraints via a post processing mechanism will be rather inefficient over an unboundedly sized structure.

In this chapter, I first review Abney's system and then present a new theory of licensing relations: *Generalized Licensing*. This system allows for a more direct connection between grammar and parser while maintaining the nice computational properties of the original system. Some constraints will, however, remain unexpressible via the new system. I therefore propose the integration of the TAG formalism with generalized licensing. The utilization of TAG allows the efficient checking of constraints since the domains we will consider will always be of bounded size - that of an TAG elementary tree domain. We will also see that the use of TAG provides us with a feasible model of incremental semantic processing. As each predicative structure is processed, it is sent off for semantic processing. Thus, we do not require a monolithic syntactic processor which completes its task entirely before the sentence is sent for interpretation. This conglomeration of TAG and generalized licensing forms the basis for the parsing mechanism which I propose. Finally, I explore the adequacy behavior of this parsing mechanism on a variety of linguistic phenomena and its implications for the analyses of these phenomena.

3.1 Abney's Licensing

In order to avoid the difficulties of the theorem proving approaches for parsing GB grammars, [Abney, 1986] abstracts away from the principles of grammar to the use of licensing relations among lexical items. Since many of the well-formedness constraints are concerned with the licensing of elements, utilizing licensing structure as a more concrete representation for parsing seems appropriate in that it allows for more efficient processing yet maintains "the spirit of the abstract grammar."

The notion of licensing which Abney endorses is one where every element in a structure must be licensed in that it performs some syntactic purpose. A structure with unlicensed elements violates this constraint and is therefore to be regarded as ill-formed.¹ This notion

¹This is quite similar to the Principle of Full Interpretation of [Chomsky, 1986b].

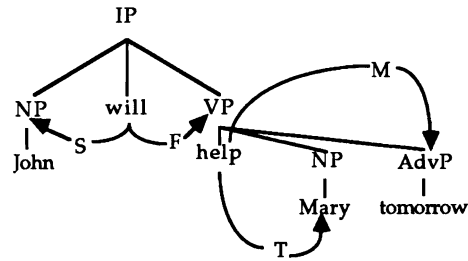
is quite reminiscent of the constraints of theta theory and case theory. The theta criterion, for example, states that every argument must be assigned a theta role, a type of licensing, or else the structure is ill-formed.

Abney takes theta role assignment to be the canonical case of licensing and thus assumes that the properties of the general licensing relation should mirror some central properties of theta assignment, namely, that theta assignment is *unique*, *local* and *lexical*. The uniqueness property for theta assignment requires that an argument receives one and only one theta role. Correspondingly, licensing is unique in that an element is licensed via exactly one licensing relation. Locality demands the theta assignment to take place under a strict definition of government: sisterhood. Sisterhood is also assumed to be the configuration under which licensing takes place. Finally, theta assignment is lexical in that it is the lexical properties of the the theta assigner which determine whether a theta assignment relation obtains and if so what that theta role will be. Again, licensing will have the same property; it is the licenser that determines how many and what sort of elements it licenses.

Now, each licensing relation is represented as a 3-tuple $\langle D, Cat, Type \rangle$. *D* is the direction in which the licensing obtains. It may be either rightward, leftward or unspecified. *Cat* is the syntactic category of the element which will be licensed by this relation. *Type* specifies the linguistic relation of which this licensing relation is an instance. This can be one of four things: *functional selection*, *subjecthood*, *modification* and *theta-assignment*. Functional selection is the relation which obtains between a functional head and the element for which it subcategorizes. Thus, this occurs between C and IP, I and VP, D and NP. Subjecthood is the relation between a head and its “subject” or in the terms of our discussion from last chapter, its specifier. Modification is the relationship between a head and an adjunct. Theta assignment occurs between a head and its subcategorized elements.

Let us examine an example of these licensing relations in a simple clause.²

²Abney’s conception of clausal phrase structure differs from the one that I have presented. Significantly, he does not assume that external arguments originate within VP. Thus, the assignment of external theta role for him is not an occurrence of licensing. This causes some problems which I will address shortly.



The I *will* licenses two things: the NP to its left is licensed via a subject relation and the VP to its right is licensed via functional selection. The V also licenses two other constituents: the object NP is licensed via theta assignment and the AdvP is licensed by modification.³ The source of all of these licensing relations is the lexicon in the entries for the individual items. The directionality of the individual relations is presumably not explicitly represented in these entries, but is instead determined by the parameters of a language.

We can now see the structural analog of licensing: the licenser and licensee are sisters and their parent is the maximal projection of the licenser. Thus, the recovery of the licensing structure is sufficient for recovering the phrase structure. Abney's parsing mechanism is then quite simple: in a single deterministic pass from left to right, assemble the licensing structure by assigning the possible licensing relations.

We can now re-examine Abney's claim that these licensing relations allow him to retain "the spirit of the abstract grammar." Since licensing relations restrict us to talk only of very local relationships, that between sisters, this system cannot enforce many other constraints such as binding, control, ECP etc. Abney admits this limitation and suggests that his work is just a "phrase structure module" of a larger system.

One would hope, though, that constraints which have their roots in licensing, such as those of theta theory and case theory, would have natural translations into this framework. Unfortunately, this is not the case. Consider, for instance the theta criterion. While this system is able to encode the portion of the constraint that requires theta roles to be assigned uniquely, it fails to guarantee the other half which requires all NPs (arguments) to receive a theta role. In fact, this crucially does not hold since NPs may also be licensed by subject licensing. The case filter, another natural licensing constraint, does not have a natural

³Note that the direction of the modification relation is slightly counter intuitive. It seems that the AdvP is modifying the VP. However, it is the V which allows the AdvP to be present in the structure and it is this facet of the relationship that licensing is intended to represent.

expression in this system since it can be independent of theta structure in ECM verbs. Thus, the following pairs will be indistinguishable with respect to licensing well-formedness:

- (32) i. It seems that the pigeon is dead
ii. Joe seems that the pigeon is dead
- (33) i. Carol asked Ben to swat the fly
ii. Carol tried Ben to swat the fly

In addition, although Abney does not address the problems of movement explicitly, we should note that standard analyses of NP movement are unavailable to his system since it fails to distinguish between subjects which receive theta roles and those which do not, as in raising or weather verbs, and also between objects of verbs which don't receive case from their theta assigner, for example ergative verbs, and objects of, for example, transitive verbs which do. Presumably, NP movement will need to be integrated into the structure building module. But, it is not at all clear how these problems can be overcome in this system. What has happened is that we have lost all sense of modularity of the various sorts of relationships about which the grammatical constraints speak. Everything is conflated onto a homogeneous licensing structure.

3.2 Licensing revisited: Generalized Licensing

Let us now consider a generalization of Abney's licensing system. There are two major problems I pointed out. One was the lack of sufficient granularity. This resulted in the inability to express the distinct sorts of relationships, case assignment and theta assignment for example, in our syntactic representation. The other problem was the inability to encode requirements on the way in which a given element needs to be licensed. This prevents us from expressing half of the theta criterion as well as all of the case filter. Clearly, these two problems are related.

To remedy these deficiencies, I propose a new system of licensing: generalized licensing. In this system, a node is assigned two sets of licensing relations. The first set is called the *gives* of that node. These are similar to the licensing relations proposed by Abney. A give is satisfied locally and is determined lexically. However, we slightly relax the assumption of uniqueness. A node may be licensed in multiple ways, possibly from the same node. We simply require the structural analogues of the licensing relations be consistent. That is, if 2 licensing relations are assigned to a given node, then there must be a well formed tree

structure in which the licensing relations both take place in their appropriate configurations. The second set will be the *needs*. These specify the ways in which a node must be licensed.⁴ Needs allow us to specify that a node is required to be licensed in a particular way. Thus, a need of type theta requires the node to be licensed by a relation of type theta. The theta criterion is thus represented by gives on a theta assigner, for example a verb, and by need feature of the type theta on all DPs. Thus, we encode both the fact that the theta role must be assigned and that the argument must receive a theta role. Needs features are usually distinguished from gives features in that they are not a property of the lexical item, but a property of some general grammatical constraint (e.g. Case filter) or as some property of the category of the projected head (e.g. modification).

All of the gives and needs will have corresponding types as in Abney's system. In this new system, though, I will allow a greater vocabulary of relation types to explicitly represent all of the assignment relations which are posited in the grammar and preserve the modularity of the theory.. Thus, we will have relations for: case, theta role assignment, modification, function selection, predication, etc. Notice now, however, that certain elements can and must be licensed redundantly. DPs, for instance, will have needs for both a theta relation and a case relation as a result of the case filter and theta criterion. Thus we will need to relax our requirement that all elements be uniquely licensed. I propose that the well formedness constraint on licensing structures is that all gives and needs features be uniquely "satisfied." The uniqueness requirement in Abney's relations has been pushed down into the level of individual gives and needs and not at the level of node. Once a give or need is satisfied, it is inert - it may not take place in any other licensing relationships.

One further generalization which I make concerns the location of these gives and needs. In Abney's system, licensing relations were associated with lexical heads and were applied to maximal projections of other heads. Phrase structure is thus entirely parasitic upon the reconstruction of the licensing relationships among the lexical items. I propose that we have an independent process of lexical projection. When an input token is received, it is projected to its maximal projection, as determined by theta structure, f-features, among other properties. This projection structure, I propose, is well-formed not as a result of any licensing relations but through this process of projection. Licensing relations are then assigned to each node of this structure. As with Abney's system of licensing, this new system will posit a structural analogue of the licensing relation. However, since we are

⁴These bear some similarity to the anti-relations of Abney. However, they are far more general.

assuming a richer system of phrase structure projection than simply a head dominated by its maximal projection, we must explicitly represent structural relations in parsing. What this gets us is that licensing relations determined by the head may now take place over a somewhat larger domain than sisterhood. We will still require licensing to occur between a node and its sister. However, we can associate licensing relations with any node in the projection of the head. Thus, the theta need which is associated with a DP as a result of the theta criterion is present only at the maximal projection. This is the node which stands in the appropriate structural relation to a potential licenser.

Let us formalize some of these notions a bit. Each node in a phrase structure tree has associated a *gives set* and a *needs set*. The gives set is composed of gives which are defined as:

A *give* is a 4-tuple $\langle D, Type, Val, Sat \rangle$ where

1. *D* is the directionality of the give, either left, right or unspecified
2. *Type* is the type of the licensing relation, an element of {Theta, Case, Function Selection, Subject, Wh, ...}
3. *Val* is the relation value, an element of $V(Type)$ for V a function which gives the possible relation values for a given licensing relation.
4. *Sat* is the node within the structure which satisfies this give. When this give is first assigned, this position is unspecified. However, upon satisfaction of the give, it is indelibly assigned.

The needs set consists of needs which are defined as:

A *need* is a 3-tuple $\langle Type, Val, Sat \rangle$ where *Type*, *Val* and *Sat* are defined as in a give.

Note that the needs are not directionally specified. This models the fact that although NPs require case and theta role they do not specify from where it should come.⁵

Now, we define what it means for a structure to be well formed.

A tree structure *t* is well-formed if and only if all of the gives and needs in the the give and need sets at each of its nodes are satisfied.

⁵We will see later in this chapter that modification might force us to adopt directionality in needs. However, the current implementation embodies the definitions I give here.

I am assuming here some formal characterization for finite tree structures. I will not offer one here. Note that this definition of tree well-formedness is similar to Mc Cawley and Peters and Ritchie interpretation of grammars as node admissibility constraints. We next define what it means for a give and a need to be satisfied.

A give $G = \langle D, Type, Val, Sat \rangle$, an element of the give set of node n in structure t , is satisfied if and only if

1. m is a node in t such that n governs m
2. m is on the side of n specified by D
3. m unifies with some element of $\mathbf{SG}(Type, Val)$ where \mathbf{SG} is a function from a relation type and value to set of partial node descriptions which correspond to the nodes which are potential licensees of the given relation and value pair

The function \mathbf{SG} might require a bit of clarification. Take for example, $\mathbf{SG}(\text{case, accusative})$. This will return a singleton set containing a description of a maximally projected DP node which has accusative case assigned to it. The descriptions which are returned by \mathbf{SG} are required to be quite small, subject to some bound, and thus the unification of each of them with any node is a constant time operation.

A need $N = \langle Type, Val, Sat \rangle$, an element of the need set of node n in structure t , is satisfied if and only if

1. Sat is a node in t and governs n
2. Sat unifies with some element of $\mathbf{SN}(Type, Val)$ where \mathbf{SN} is a function from a relation type and value to set of partial node descriptions which correspond to the nodes which are potential licensers of the given relation and value pair

Finally, we need a restricted definition for government.

A node n governs a node m in a structure t if and only if

1. n is sister to m in t
2. the node k which immediately dominates m and n is a projection of n

Let us consider a concrete example of all of this. Particularly, let us examine the following structure and see what the relationships among the nodes are.

(34) $[V'_j [\text{detest} [DP_k \text{ the } [N'_l \text{ man }]]]]$

First, what are the gives and needs associated with each of the nodes prior to any satisfaction? The lexical head dominating *detest* has 2 gives: $\langle \rightarrow, \text{case, accusative, ?} \rangle$ and $\langle \rightarrow, \text{theta, D, ?} \rangle$.⁶ Node *k*, the DP node, has associated with it 2 needs: $\langle \text{case, ?, ?} \rangle$ and $\langle \text{theta, ?, ?} \rangle$. Each of these gives and needs are satisfiable by the other node since the V^0 node governs the DP node and the SG and SN functions return appropriate descriptions. The D^0 node's give set consists of single give: $\langle \rightarrow, \text{function-select, N, ?} \rangle$. In this case, the N' node dominating *man*, node *l*, is appropriate and serves to satisfy this requirement.

The reader may have noticed that the current formulation of licensing will often result in a node not being able to have all of its needs satisfied in a single position. In a sense, it is required to be in two different structural positions at once. Consider the case of passive. The subject of the sentence is said to receive its theta role from the verb (as its internal argument), yet it receives its case from the tense/agreement morpheme on the verb. Clearly, it cannot do both of these. Given the structural correlate of the licensing relation, the verb would have to be directly dominated both by IP and by VP. But this is impossible in a tree structure. Yet if we assign it to either position we will have an ill formed structure since we will have (at least) unsatisfied needs. Thus, our representation and constraints on give and need satisfaction have forced us into adopting a notion of chain. A chain will consist of a list of nodes $\langle a_1, \dots, a_n \rangle$ such that they share all gives and needs and each element c-commands the following. The first element in the chain, a_1 , the head of the chain, is the only element which is permitted to possess phonological content. All of the remainder are empty categories.

Now, since the elements of the chain are distinct nodes, they can occupy different structural positions and thus be governed by distinct heads. In the passive sentence:

(35) $[IP \text{ John}_i \text{ was } [V' \text{ killed } t_i]]]$

The trace node which is complement of *killed* forms a chain with the DP *John*. In its V' internal position, the theta need is satisfied. It is governed by the verb and satisfies the

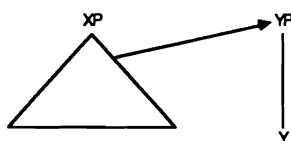
⁶Unspecified values are given as ?. Also, theta gives and needs have as their value the category of the node which will receive this theta role. Of course, we will somehow need to include in this value the semantic role associated with this theta role assignment.

theta give associated with the zero level projection of V. In subject position, it receives case from (and is governed by) the I' projection of the tns/agr morphology.

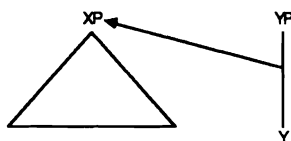
3.3 Using Generalized Licensing for Parsing

In the previous section, I proposed a system of generalized licensing relations which provide constraints on well formedness of syntactic structures. How might we build a parser which builds structures in accordance with these relations? Also, how might such an algorithm be constructed such that it remains efficient and psychologically plausible? Finally, once these licensing relation structures are constructed how may other constraints not enforced through licensing be checked in an efficient manner?

I shall propose a parsing mechanism which is quite simple. It proceeds one token at a time, first projecting a token, adding its associated gives and needs, and attempting to combine it with previously built structure in one of two ways: The first is to attach it as the sister of some node on the right frontier of the previously built structure as a result of a give on the right frontier which the root of the projected structure satisfies. The following figure illustrates this:



Another possibility is that the previously built structure is attached on the left frontier of the projected structure thereby satisfying a give on one the projection's nodes.



Often times there will be many possible attachment sites. In such cases, we will have an ordering of relations which will choose one of the possibilities. This heuristic is intended to model human attachment preferences and will certainly be influenced by such factors as intonation and pragmatics as well as the syntactic preferences. I will discuss the specifics of this ordering a bit later.

Now, as structure is built up, certain gives and needs will no longer be exposed to the

right frontier of the working structure. If any of their gives and needs remain unsatisfied, they will be unable to do so in this position. In the case of a need which is unsatisfied in node n , we can posit the existence of another node m , which will be attached later to the structure such that $\langle n, m \rangle$ form a chain. That is we posit an element to have been moved into its structural position exactly when it is licensed in that position, but its needs are not completely satisfied. When we posit such an element, we push it onto the *trace stack*. As other nodes are encountered which no longer have access to the right frontier yet have remaining needs, they are pushed onto the stack as well. When a node has a give which remains unsatisfied and the node no longer has access to the right frontier, we know that if this string is to be well formed, there must be some element, not phonologically represented in the input list of tokens such that it satisfies that give relation. At this point, we check whether the trace on top of the trace stack would satisfy this give. If this is the case, we pop it off the stack and attach it. Of course if it has any remaining needs, we push it back onto the trace stack since it will no longer have access to the right frontier. If however, either no elements appear on the trace stack or the element on top of the stack will not satisfy the give, we check whether there is an empty element which will satisfy the appropriate give.⁷

Consider a simple example. Suppose we are processing the sentence “John laughs.” The first token the parser receives is *John*. This is projected to DP. No gives are associated with this node, but we know that all phonologically overt DPs have case and theta needs from the theta criterion and the case filter. We insert them into the need set of the DP node and continue processing. Our next input token is *tns/agr*.⁸ This is of type I and projects to two bar levels, since this particular I possesses f-features. Associated with the I node is a rightward give of function selection of value V. Associated with the I' node is a leftward give of nominative case and a leftward give of subject, as a result of the Extended Projection Principle. We then attach the DP as sister to the I' node thereby satisfying the subject and case gives of the I' as well as the case need of the DP. However, the theta need remains. Thus, we posit an empty category which is of type DP and possesses (only) a theta need and push it on the stack. The next input token is the verb *laugh*. This is projected to a

⁷Note that the use of this stack to recover dependencies guarantees that we will not allow nested dependencies as was done in [Fodor, 1978]. With this natural computational mechanism, then, we are able to enforce at least part the Path Containment Constraint of [Pesetsky, 1982]. The slight hedging here results from our lack of a INFL to COMP path on which much of Pesetsky's analysis rests. However, such a path might be integrated into the mechanism.

⁸I am assuming the input to the parser to be the output of some morphological pre-processor. Of course, it is not this simple since there is ambiguity between bare verbs and their inflected forms: The men (*tns/agr*) die vs. The men want to die

single bar level. Since *laugh* assigns only a single external theta role, we insert the only give of this projection to the V' node - a leftward theta give to a category of type DP. This verbal projection is attached as daughter to I' - it then satisfies the function-selection give of I . However, the theta give in V' is unsatisfied and since it is leftward and is inaccessible from the right frontier. We see that we need to posit an empty category to satisfy this give. We see that there is a DP trace on top of the trace stack which will accept this give, for recall that in subject position the DP was assigned only case and not theta role. This trace is popped off of the stack and is attached via chomsky-adjunction as sister to the V' node. Since this node forms a chain with the subject DP, the theta need on the subject DP is now satisfied. We have now reached the end of our input. The resulting structure is easily seen to be well formed: all gives and needs are satisfied. Additionally, the trace stack is empty, a requirement we add to the well-formedness of a structure.⁹

We have adopted a very particular view of traces: their positions in the structure must be independently motivated by some other licensing relation. So, all movement must be strictly structure preserving. Note that we cannot analyze long distance dependencies through successive cyclic movement. There is no licensing relation which will cause the intermediate traces to exist. Ordinarily these traces exist only to allow a well-formed derivation, i.e. not ruled out by subjacency or a barrier to antecedent government. Thus, we will have to account for constraints on long distance movement in another manner. We will return to this in section 3.5

3.4 Encoding of Grammar through Generalized Licensing

One of the criticisms of Abney's licensing relations which I presented above was that they cannot fully encode all of the sorts of constraints which our theory of grammar might posit. Our new system of licensing, although slightly more refined and expressive, will still not allow us to state as large a class of constraints as we might like. The only dependencies which can be captured remain those among sister nodes in the tree and between a node and those nodes in its projection.

My proposal, then, is to explicitly divide grammatical constraints into two classes: those expressible with licensing (and thereby provide structure building information) and those

⁹Actually this is redundant with the requirement that all needs be satisfied since if they were not satisfied, they would be on the trace stack with the possible exception of a need which was accessible to the right frontier throughout its lifetime in the derivation

I shall characterize as abstract well-formedness checks. The division of labor between a constrained feature checking system which is used “on-line” during processing and a more powerful constraint satisfaction system performed after structures are constructed will, I believe, result in a more efficient realization of a principles and parameters form of grammar. Now, the immediate problem to be solved in this approach is deciding which class of constraints a given principle falls into.

Recall that Abney’s original licensing relations were designed as a generalization of theta role assignment. We should then expect that the constraints of theta theory would be storable in the language of generalized licensing. This is, in fact, the case. Recall from chapter 2 that the theta criterion states that:

- (36) i. Every argument must be assigned (exactly) one theta role.
ii. Every theta role must be assigned to (exactly) one argument.

The first clause is easily encoded by assigning a need feature of type theta to all argument maximal projections.¹⁰ If a structure is produced in which this phrase is not assigned a theta role, it is deemed ungrammatical since this structure will have an unsatisfied need and is therefore not well formed. The second clause is enforced through gives features associated with the theta role assigner. The status of a lexical item as a theta role assigner is part of that item’s lexical entry. In the system, we have theta grids associated with each of the theta role assigning items. One element may be distinguished as the external argument. The directionality of these gives is derived from general directionality facts about the language. Internal theta role assignment takes place in the direction of complements in a language. External theta role assignment takes place in the direction of specifiers. These directions are specified as part of the parameters for the language to be parsed. The location of these gives is determined by the type of theta role assignment. For internal theta role assignment, the give is associated with the V^0 projection. External theta role assignment by definition takes place at the V' level. Now, the uniqueness of the assignment comes as a result of the

¹⁰The definition of argument maximal projection is not entirely obvious. A CP, for example, can function as an adverbial adjunct, a relative clause or a complement. DPs can function as arguments, predicate expressions or appositives. This raises a problem, then, for the assignment of features immediately upon projection. However, it does not seem insurmountable. One possible solution might be to assign theta needs in all of these cases but loosen the definition of well formed structures thereby allowing, for example, DPs which have not been assigned theta roles but are functioning as predicate nominals. Another possibility for future exploration is the use of disjunctive constraints. One problem in using such devices, though, is their indirect relation to the (abstract) grammar. There is no principle in the grammar which states that a DP can function in the manner of x, y or z. Thus, the use of disjunction will entail some compilation of the grammatical principles.

process of give satisfaction. In general, this sort of bijective relation between licensee need and licenser give is the strongest requirement that licensing can easily encode.

Another example of a constraint which requires such a bijective relation is case assignment. The case filter stipulates that a lexically realized noun phrase must possess (abstract) case. Additionally, the case must be assigned uniquely. Again, this is easily encoded by a need feature on the maximal projection of all DPs. The strict locality and uniqueness of case assignment is expressed through the constraints on the satisfaction of the case gives associated with the case assigning node.¹¹ As with theta roles, the case assigning properties of a lexical item are given in the lexicon in the form of a case grid. The directionality of canonical case assignment is given as a language specific parameter. Case assigned by a functional category is determined by the presence of f-features in the lexical entry. Possible values for f-features are genitive and nominative case. In these situations, we project the functional category to two bar levels and add the give appropriate to the f-features to the the X' level. The directionality of this give is clearly determined by specifier directionality.

Other principles may also be expressed through licensing. If we assume that the notion of subject is a structural one, than we may express the extension to the projection principle, the requirement that all clauses have subjects, through a give associated with the first projection of I in the direction of specifiers for the language. Subcategorization information of functional categories, a property lexically specified for the functional categories in general and not for the particular functional lexical items, will also be expressed through licensing. This *functional selection* is accomplished by a give of type function-select with value of the appropriate node type associated with the functional head in the complement direction. Finally, we shall also encode modification through the use of licensing. Notice that we will need to associate a modification need with the maximal projection of the modifier structure. This need will be satisfiable by a constrained set of maximal projections depending upon the category of the modification structure. In this case and the functional selection case, the

¹¹It has been proposed that some cases of case assignment are optional. The assignment of so-called exceptional case by verbs such as “want” apparently does not always take place:

- i. Alvin wanted [_{IP} the chipmunks to sing]
- ii. Alvin wanted [_{IP} PRO to sing]

In i, *the chipmunks* are standardly assumed to receive case from the verb *want* and are thus not ruled out by the case filter. In ii, however, PRO must not receive case since case assignment is assumed to take place under government and PRO is not permitted to be governed (by the PRO theorem). Thus, we may need some additional mechanisms for allowing gives associated with assignment of case to be unsatisfied in well-formed structures. Alternatively, we can allow want to be truly ambiguous between a case assigning and non-case assigning version.

need and give are not reciprocated. That is, we do not see a give and need simultaneously satisfying one another. When, for example a VP satisfies the function-select give associated with an I, there is no need on the VP which is in turn satisfied. Likewise, when the modification need is satisfied, no give associated with the modified maximal projection is satisfied. This gives us the power to represent the optionality of certain attachments. We should not require that, for example, a DP is always modified. Thus, unless we are going to allow well formed structures which possess unsatisfied gives, we will not be able to determine whether to associate the give feature with the DP. Of course, our system of licensing relations does not require that all relations be reciprocated. Recall that the requirements on satisfaction of a give or need were simply that they be consistent with some element of the set returned as the value of SG or SN.

Next, let us consider the constraints which have not yet been discussed as instances of licensing. Consider the case of anaphor binding. Binding theory principle A states that an anaphor must be bound in its governing category. Could this be expressed through the use of licensing? What sort of need feature could we give to the projection of, say, *himself* which would guarantee that it is bound in the proper domain? Perhaps we could have some sort of feature passing such that the gives and needs of the verbal projection are altered once an anaphor is attached. Or else we could have many different projections for a given verb - one which required an anaphor object and a coindexed subject. Not only would these require additional machinery to be added to our current system, they seem out of the spirit of our enterprise. Neither of these “fixes” gives us the power to state truly abstract constraints, but rather adds a mechanism for encoding a class of dependency which I believe is much more than we need.

Like binding theory, many other principles of GB are impossible to express perspicuously as instances of licensing. These include the ECP, control theory and bounding theory (i.e. subjacency). Remember that we still want our parser to maintain a fairly direct relationship with the grammar. Thus, we will need some mechanism by which we can explicitly check that all of the principles are satisfied in our final structures. Checking in a “brute force” manner that an arbitrary structure satisfies a set of first order formulae (the principles of our theory) is unfortunately quite inefficient. The problem of assigning a grammatical set of indices which satisfies binding theory can be shown to be NP-hard.¹² Since we want our parser to behave in a psychologically plausible manner, we cannot allow such inefficiency

¹²Robert Berwick, p.c.

to creep into our system. Another breach of psychological plausibility comes from when we check these constraints and thus when we can determine a sentence to be ungrammatical. People are apparently able to determine a sentence to be ungrammatical as soon as it is possible to do so.¹³ Let us call this temporal locality of constraint checking. If we wait until the entire sentence is parsed before we check all of our constraints, we will not maintain this temporal locality. However, if we check constraints after each word is integrated into our structure, thereby maintaining the property, we introduce even more processing complexity. Additionally, it is not even clear that we will be able to determine at what stage a given constraint ought to apply. We do not want to rule a structure ungrammatical simply because it is not yet complete.

The question, then, is whether we can maintain temporal locality of constraint checking and processing efficiency. The answer is yes. This will be achieved through the use of Tree Adjoining Grammar as the formalism over which our parser will operate. The manner in which this is done is discussed in the next section.

3.5 Limiting the Domain using TAG

At the end of the previous section, we raised the question of how we might preserve processing efficiency and temporal locality of constraint checking. We noted that the checking of satisfaction of a set of first order formulae in a structure of unbounded size is quite difficult. Yet, we do want to maintain the direct link between the principles of grammar and the operation of the parser. One way of reconciling these two hopes is by bounding the size of the structure over which such constraints need to be checked. However, linguistic structures are allowed to be of unbounded size. There seems to be a contradiction, then.

Tree adjoining grammar provides us with a way out of this dilemma. Recall from chapter 2 that TAG accomplishes its linguistic description by factoring out recursion from local dependencies. Suppose that we can guarantee that all of the principles of the grammar are storable over a single local domain as determined by a TAG elementary tree. Then we can use the parser to construct a set of elementary trees. Once a tree is finished, it is checked for satisfaction of all relevant constraints. Since each of these trees is of bounded size, our constraint checking will be of constant time and thus we will retain efficient processing.¹⁴

¹³This has led some to posit LR parsing, which possesses this property among others, as a model for human language processing [Shieber, 1983].

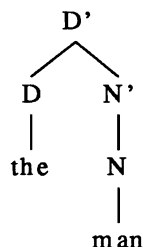
¹⁴A real psychological model of parsing will certainly not explicitly perform constraint checking. Thus,

Additionally, we will have temporal locality of constraint checking as well since constraints will be checked as soon as they may be - that is once the minimal structure over which they are stated is complete.

Remember that this proposal depends on a rather strong empirical claim: that all of the principles of grammar are expressible either through licensing relations or as constraints over such local domains. In this work, I do not attempt in any way to “prove” this claim. However, it does seem that as far as it has been investigated, such a formulation of grammar is possible.

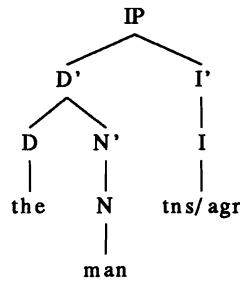
We need now to re-evaluate our “parsing using licensing relations” story from the perspective of TAG elementary tree construction. The basic idea is this: The parser will operate as before, projecting the input tokens, attaching them, positing empty categories and so forth. However, we will constrain the parser’s *working space* such that at any time it may only deal with a structure corresponding to a single elementary tree. Recall from the previous chapter that an elementary tree corresponds to a single predication. As soon as we perform an attachment which violates our “memory limitations”, we will be forced to reduce the structure in our working space. We will do this in one of two ways, corresponding to the two mechanisms which TAG provides for building structure. Either we will undo a substitution or undo an adjunction.

Suppose, for instance, that we are parsing the sentence “The man hiccuped”. After parsing the input “the man” we will have built the structure:



The next input token we receive is the tense and agreement morphology (tns/agr). Being of category I, this projects to IP (= S). The appropriate licensing features are added and the DP is attached as the spec of IP as a result of the case assigning and subject licensing.

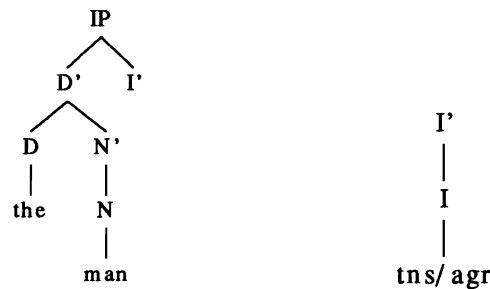
we might imagine computational processes which perform the analogs of guaranteeing that each of the constraints are satisfied. Nonetheless, preserving locality of the the domains over which these computations take place will be necessary.



We are left with a problem, though. The D' constitutes a predication since it contains the projection of a lexical head, in this case N, and all of its associated functional projections. In addition, the IP determines another predication, one which will ultimately have as its lexical head a verb. Thus, we must reduce the size of our working space. In this case, we excise the DP predication from the structure by undoing a substitution. We get the following two elementary trees:



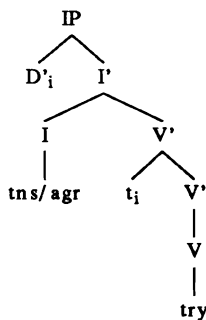
Note that one other operation could have been performed to reduce the size of our processing domain. We could have unadjoined the subject from the structure using the two projections of IP as the recursive root and foot nodes and thereby yielding the following elementary trees:



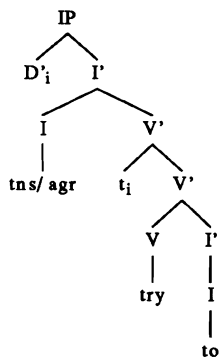
There is one problem with this analysis. Although we do not require that licensing take place within a single elementary tree, we do require that all dependencies reside within a local domain. This is a core property of TAG. Now, note that upon attachment to the IP, the case need is satisfied, but the theta need is not. Thus, we place the DP on the trace stack, waiting for a spot in which it can function as a gap. Presumably, this would happen as the verb's external argument and would thus land in a position external to the

subject adjunct elementary tree. If we unadjoin the subject, then we will not have the position in which it receives a theta role within the same structure as the position in which it receives case. Alternatively, we can state that upon removal from the working structure, an elementary tree needs to have all of its licensing relations (both gives and needs) satisfied. This is sufficient since once a structure is excised, there is no chance for remaining input to saturate any of its licensing relations.¹⁵

As a second example of the processing domain reduction process, consider how we will handle the sentence “The fool tried to be king”. After parsing the input “The fool tried”¹⁶ we have:¹⁷



We project the next input token, *to*, add its licensing features, and perform the appropriate attachment. Thus, we are left with:



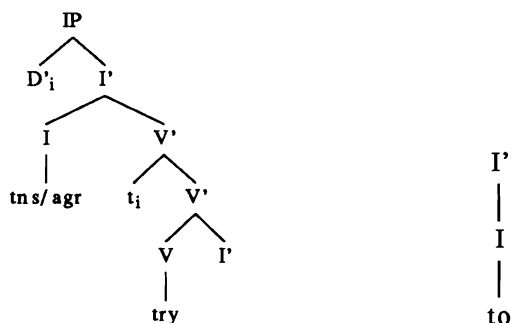
Now, at this point, we have projections corresponding to two distinct predications present in the working space: the matrix predication, containing the lexical projection

¹⁵Actually, this is a bit of a simplification. It depends upon the relative position of the excised structure to main working space structure. If the excised structure is on the right frontier and is not satisfied, as in prepositional phrases, we will temporarily focus our attention on the excised structure as our main processing structure. Thus we will need to associate trace stacks with each potential structure to be placed in a processing domain (i.e. a stack of stacks) We will consider such cases shortly.

¹⁶Including, of course, the token for tns/agr.

¹⁷Note that the DP tree corresponding to *The fool* has already been excised by un-substitution as was described in the previous example.

of V and its associated functional projections (in this case I), as well as the subordinate predication, represented by the projection of I. Here, we undo an adjunction. The recursive structure which we excise is that between the tree root and the complement of *try*:



In this case, there are no remaining unsatisfied gives and needs in the structure of the matrix sentence and thus the unadjunction was licit. Note that the unsubstitution of the complement IP would also have been acceptable. In fact, the resulting set of structures would be identical, modulo the inheritance of features of the root node in the new working space structure. However, I shall take as a general heuristic that if an unadjunction is possible, then that will be performed. Otherwise, we will perform an unsubstitution.

To recapitulate, using these sorts of processing domain reductions, we are able to maintain a small structure in the parser's working space. Therefore, at any time, we need only consider a bounded number of attachment possibilities. In addition, upon excision from the current structure, a newly created elementary tree will be appropriate for constraint checking and semantic interpretation.

We are now ready to state the complete parsing algorithm:

1. **Projection:** Read input token and project with relevant gives and needs assigned to all nodes of the projection (PROJ). Check working space structure (WSS). If WSS is empty, go to step 1. Otherwise, continue.
2. **Attachment:**
 - (a) Determine all licensing possibilities between the PROJ and WSS. These are either between the maximal projection of PROJ and some node on the right frontier¹⁸ of WSS or between some node on the left frontier of PROJ and the root of WSS.

¹⁸If the root node of the WSS is the site of a previous unadjunction, we cannot allow a node in PROJ to attach by licensing this root. This will violate previous attachments. This is easily enforceable through the use of a flag associated with such a node upon the unadjunction.

- (b) If there are no attachment possibilities, posit empty head which will satisfy some of the unsatisfied gives on the right frontier of the wss and return to 1. Otherwise continue.
- (c) Order this list of licensing possibilities based upon type of licensing relation and relative depth in the tree. The ordering based upon type of relation is: functional selection > case > theta > subject > modification. Within each of these, low attachment is preferred.
- (d) Perform attachment and constraint satisfaction associated with relevant licensing relations.

3. **Domain Reduction:** Check size of wss domain. If only a single predication, go to step 4. Otherwise,

- (a) Determine whether recursive structure exists such that an unadjunction may be performed and check that all gives and needs are satisfied within the domain of this structure (i.e. all dependencies are localized) with the exception of gives on the right frontier. If one exists, undo adjunction. Otherwise, perform relevant un-substitution. Call the newly unadjoined or unsubstituted structure **US**.
- (b) For **US**, check if right frontier is satisfied. If so, pass this structure to constraint checking and semantic processing mechanism and go to step 4. If right frontier is not satisfied, check if nodes on top of trace stack are within new structure and will satisfy these gives. If so, attach these traces, send **US** to constraint checking and semantic processing, and proceed to step 4. Otherwise, push wss and trace stack onto context stack. Set wss to be **US** and set the trace stack to be those elements on the top of the old trace stack which were within **US**.

4. **Resolution of unsatisfied gives and needs:**

needs If the wss is attached such that its root is no longer on the right frontier, check whether its root possesses any unsatisfied needs. If yes, push the node onto the trace stack.

gives For any unsatisfied rightward gives which are not on the right frontier of the wss or any leftward gives which have not been satisfied by attachment, we will posit and attach empty categories. This process proceeds bottom up. The empty category is of type trace if the node on top of the trace stack is of correct type - as

determined by the function **SG** - and the trace stack is then popped. Otherwise, the empty category will be a generic empty category of appropriate syntactic category. Note that once the empty category is attached, its unsatisfied gives and needs - as determined in the projection process - must also be assessed. Alternatively, if an internal node is made accessible to an internal leftward give as a result of an unadjunction, this can be used instead of an empty category to satisfy this give.

5. If the context stack is non-empty and the WSS is completely satisfied with respect to gives and needs, we pop the context stack restoring the old WSS and trace stack. This step is repeated until either the context stack is empty or else some WSS is unsaturated.
6. Go to step 1.

3.6 Linguistic Adequacy and Implications of the Parsing Mechanism

In this section, we will carefully consider how the proposed parser fares on various linguistic phenomena. In the first example, I will go through the parser's operations in gory detail so that the operation is entirely clear. Thereafter, I will note only the interesting "landmarks" along the way.

3.6.1 Raising

Raising sentences are distinguished by the fact that their surface subject is semantically associated with the lower clause. It is thus said to have been "raised." The example we will consider in this section is:

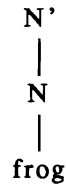
(37) The frog seemed to kiss Cinderella.

Initially, the parser's working space and trace stack are both empty. The first token, *the*, will be projected to the following structure:

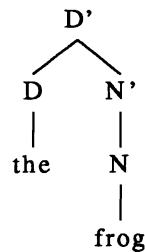
$$\begin{array}{c} D' \\ | \\ D \\ | \\ \text{the} \end{array}$$

Associated with the D^0 projection is a function select give of value N to the right. The D' maximal projection will have a need set with a need of case and of theta-role. Unless explicitly mentioned, all other give and need sets are taken to empty. Since the wss is empty, we put this projection in the wss and continue.

We now receive the input *frog* which projects to:

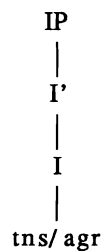


This projection has no associated gives or needs. We check all possible attachment sites along the right frontier of the wss for the N' and find that it is licensed as sister to the D^0 as a result of the function-select give. No other licensing relations exist along this frontier nor do they exist on the left frontier of the N' projection to license the wss. Thus, we perform this attachment and thereby satisfy the relevant give yielding:



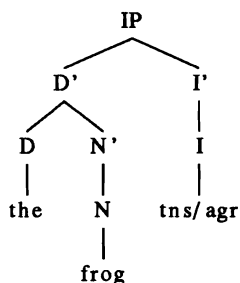
We must now check whether the wss domain consists of only a single predication. In this case, it does, so we need not perform any domain reduction operations. We need not consider any of the unsatisfied needs since the wss root remains on the right frontier. Additionally, there are no tree internal gives which are unsatisfied. Thus, we do not posit any empty categories and the trace stack remains empty.

Continuing on, the next input token is *tns/agr* which projects to:

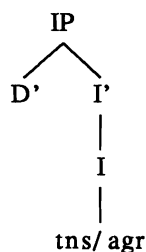


Associated with the I^0 , we have a rightward function select give of value V. The I' node has 2 leftward gives - one is a case give of value nominative, the other is a subject give. Again we check for all licensing possibilities and find a unique possibility - that of the I' licensing

the DP projection - thereby satisfying the case need on the D' and the case and subject gives on the I'. We perform this attachment and get:



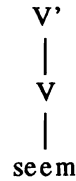
At this point we check for multiple predications and find that we have two: that determined by the N lexical head and its associated functional projections and that in which the projection of I will ultimately reside. However, we find that no recursive structure exists and thus must perform an unsubstitution - in this case of the subtree rooted at the D' node. No unsatisfied gives or needs exist in this subtree. Note that we do not consider the unsatisfied theta need which is associated with the root of the DP since this need will be represented by the substitution node which remains in the WSS. Thus, we send off the unsubstituted DP tree to further constraint checking and semantic processing and continue on with this structure in our working space:



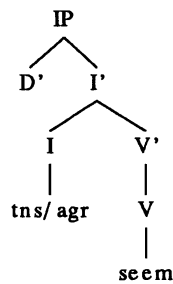
Of course, we do record the information about where the domain reduction took place so that we may later perform compositional semantics on the elementary tree structures.

In the attachment which we just performed, the WSS (now represented as the DP substitution site) was attached to the new projection in such a way as to remove its root from the right frontier. In addition, this root had an unsatisfied theta need. Thus, we place a copy of this node (which will later serve as a theta receiving node forming a chain with the original node) onto the trace stack. There are no unsatisfied internal gives and we therefore proceed with the next token.

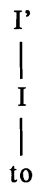
The verb *seem* is projected as:



The only give associated with it is a rightward theta give of value I on the V^0 node. No needs are added to the structure. Once more, we check for possible attachment sites and find a unique possible licensing relation between the function select give of the I and the V' and this attachment is thus performed yielding:

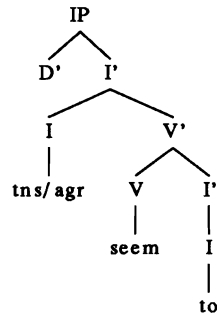


Here we do not have multiple predications in our wss and do not need to perform any domain reduction.¹⁹ The root of the wss remains on the right frontier and we therefore do not need to push any nodes onto the trace stack. Also, there are no internal unsatisfied gives so we continue with the next input token, *to*. This projects to:



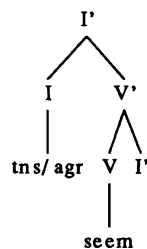
Associated with the I^0 node is a rightward function select give of value V. The I' node has associated with it a leftward subject give - as a result of the extended projection principle. This time, we check for possible attachments and find that there are two. The first is having the wss satisfy the subject give of the I' , assuming, of course that IPs can do such things. The second is having the newly projected I' satisfy the theta give of the V^0 . Our ordering of attachment preferences specifies that theta requirements are in some sense “stronger” than subject requirements and we therefore perform the attachment of the new projection as sister of the V^0 in the wss.

¹⁹Crucially, the foot nodes which result from unsubstitution do not count as determining a distinct predication

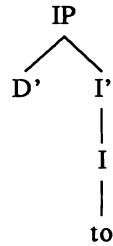


We now observe that the wss once again is comprised of two distinct predications, one being the matrix sentence (i.e. the projection of the verbal lexical head with its associated functional projections, the IP in this case) and the other being the lower clause determined by the recently projected I. In this case, there are two recursive structures that could be unadjoined. The structure which lies between the root node of the wss and the maximal projection of the lower I, cannot be removed though. The specifier of the matrix IP, the DP substitution site, remains in the trace stack and has not received a theta role. If this structure were removed, a chain containing the DP could never receive theta role since all chains must be internal to elementary trees - in violation of the theta criterion. Alternatively, we can think about this as a restriction against removing in a domain reduction operation any node residing in the trace stack. This is equivalent since a node goes into the trace stack so that it can become part of a chain which will satisfy a need which was not satisfied in-situ.

In contrast, the structure which lies between the I' of the matrix sentence (i.e. the daughter of the wss root) and the I' of the new projection can be unadjoined since the DP node will remain in the wss and will therefore be eligible for reintroduction as a trace in the structure yet to be built. Thus, we remove the structure:

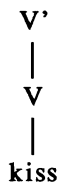


send it off for constraint checking and semantic interpretation and continue with the following as the wss:

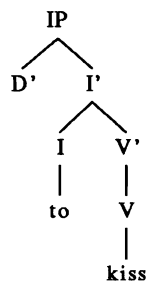


We do not need to worry about unsatisfied needs, but note that there is an internal unsatisfied give: the leftward subject give of the I' . Note, though, that we do not need to posit the existence of an empty category here. Rather, the unadjunction of the “seemed” tree has made the DP node accessible as the possible satisfier of this give. We therefore satisfy this give. Notice here that the ordering of the domain reduction process with respect to the resolution of unsatisfied gives and needs is crucial. If we were to have posited an empty category to satisfy the subject give immediately upon attachment of the new I projection, we would not have had the recursive structure available for unsubstitution.

At this point, we have no outstanding unsatisfied gives and needs which need to be immediately resolved so we proceed to the next input token *kiss*. This verb projects to the following:

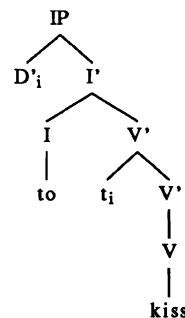


Associated with the V^0 are 2 rightward gives, one is a theta give of value D and the other is a case give of value accusative. The V' has a single give, a leftward theta give of value D. The only possible licensing relation is that between the function select give on the I^0 and the V' . This is attached yielding the structure:

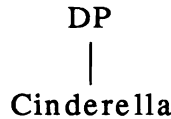


At this point our WSS consists of a single predication and we therefore do not need to unadjoin or unsubstitute. And since the root of the WSS remains so after the attachment, we do not need to resolve any unsatisfied needs. However, the leftward theta give associated

with the V' is inaccessible from the right frontier and must be satisfied. Now we check the trace stack for an element of the appropriate type to satisfy this give. In fact, there is such an element there, a copy of the DP node in the specifier of I. This node is attached as sister to the V' - forcing a chomsky adjunction - and both the theta give on the V' and the theta need on the DP are satisfied. This DP possesses no more unsatisfied needs so we are not required to remove it and push it back on the trace stack.²⁰ We are left with this in our WSS:



Finally, we receive the token *Cinderella* which is lexically specified as a full DP. Its simple projection is:

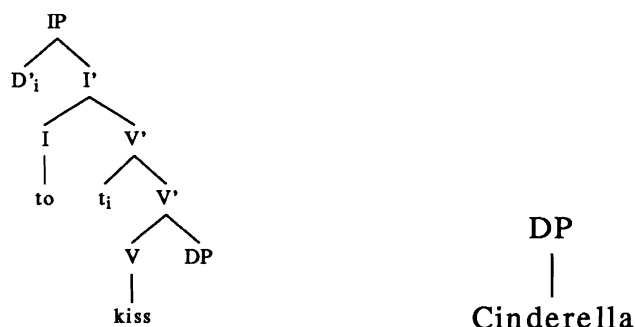


On the DP node are 2 needs: theta and case. Again, we find a unique attachment possibility - as the object of kiss - which satisfies all of the remaining gives and needs. The DP is immediately seen to function as a second predication instance, so we must perform some domain reduction. This is accomplished by unsubstitution since there is no appropriate recursive structure for unadjunction. No unsatisfied gives or needs remain and the trace stack is empty so we are finished.

In sum, we produced the following set of elementary structures:



²⁰As with the substitution and foot nodes, we stipulate that traces and other empty categories do not constitute predications which need to be unsubstituted from an elementary tree.



The derivation structure which represents the methods of combination of the various structures has also been constructed and may thus be used in semantic interpretation.²¹

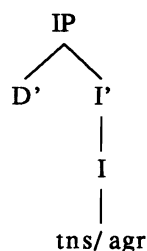
3.6.2 Control

Now consider the case of a control verb, such as *try*, in the sentence:

(38) The frog tried to kiss Cinderella.

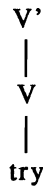
This sentence forms a sort of minimal pair with the raising case considered above. Both assign internal theta roles to constituents of type IP. However, while the verb *seem* assigned no theta role to an external argument, *try* assigns an agent external theta role to a constituent of category DP.

Let us see how this minimal lexical difference affects the processing of the sentence. Obviously, on the input, *The frog tns/agr*, the behavior will be identical in both cases. On the trace stack will be a node which is a copy of the subject DP which requires a theta role. The context stack will be empty. The wss is the following:

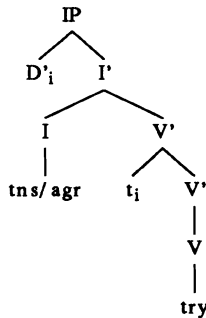


We next receive the verb *try* which is projected to:

²¹It would be interesting to investigate how incremental semantic interpretation on a derivation structure might be performed

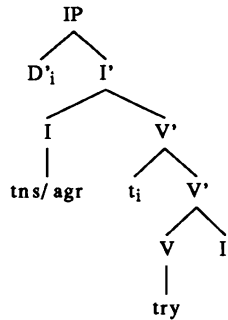


Associated with the V^0 node is a rightward theta give of type I. This is identical to the *seem* projection. However, associated with the V' node is a leftward theta give of type D. The only possible licensing relation is again determined to be the function selection give on the I^0 node to the newly projected V' node and the attachment is performed. No domain reduction need be performed, so we continue on to resolve unsatisfied gives and needs. Since the root of the wss remains the root after that attachment, we don't worry about unsatisfied needs. There is, however, an internal give which needs to be satisfied: the leftward theta give on the V' node. We see that the empty category on the trace stack is a good candidate for the receipt of this give and therefore pop the trace stack and attach it as sister to the V' through Chomsky adjunction giving the structure:



This attachment has also satisfied the empty category's theta need and it therefore need not be pushed back onto the trace stack.

Continuing as before, *to* is projected to IP with a rightward function select give of type V on the I^0 node and a leftward subject give on the I' node. This is attached as sister to the V as a result of the theta give. We now proceed with domain reduction. This time, though, there is an important difference: there are no nodes in the higher IP projection which remain unsatisfied. The subject DP (i.e. the specifier of I) has received its theta role as external argument of *try*. Therefore, we are able to remove the entire upper IP as a recursive structure yielding the following auxiliary structure:



All that remains in the wss is the simple projection of the lexical item *to*. One might wonder whether we are forced to appeal to some “maximality” principle²² in order to rule out the previously exploited unadjunction possibility. However, this is not the case. This unadjunction is not possible since the specifier needs to be in the same elementary tree as all nodes with which it enters into a chain relation. Thus, the assignment of the external theta role of the verb forces us into the correct possibility. The property of removing theta chains in a single domain reduction guarantees us that a category will be semantically interpreted as soon as it may be. Since domain reductions are centered around a structure determined by a lexical head (i.e. an argument taking function) and we require that all arguments appear in the projection determined by the lexical head, the removal of the lexical head will entail the removal of all and only the relevant theta chains.

We continue by checking unsatisfied gives and needs in the wss. The leftward subject give on I' is not satisfied and is not accessible from the right frontier. Since the trace stack is empty, we posit an empty category of the appropriate type for this give in this case PRO. This empty category is projected and attached in the appropriate position. The DP node dominating PRO has one unsatisfied need requiring a theta role. Since this is not satisfiable in the current position, we place a copy of this node on the trace stack.

We now continue with the remainder of the sentence in exactly the same manner as the raising case. We attach the verb, pop the trace off of the trace stack, this time it is the trace of PRO instead of the matrix clause subject, to satisfy the external argument theta give, attach the object, and so forth.

3.6.3 Exceptional Case Marking and Anaphor Binding

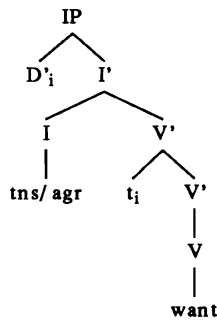
Exceptional case marking is exemplified in the following sentence:

²²That is, “remove as big a piece of structure as you possibly can.”

(39) The astrologer wanted Nancy to pay.

The exceptional case marking verb *want* is lexically specified as assigning an external theta role to an element of type DP, an internal theta role to an element of type IP, and accusative case. Thus, we have another minimal pair consisting of this verb and the control verb, *try*, which has identical lexical information with the omission of the case assignment.

The processing of the initial segment, *The astrologer tns/agr want*, will be identical to that shown above for the control case. The trace stack is empty and the wss is:



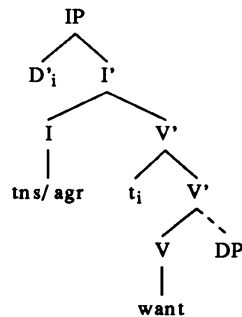
The V^0 node will have a rightward theta give as before but will also have a rightward case give. When the next token, *Nancy*, is received, it is projected to DP which has associated with it theta and case needs. The only licensing possibility is between the V^0 node and the DP thereby satisfying both the case give and need. Note, though, that attaching the DP as sister of V is not really correct. If we want constituency to directly reflect argument structure, then the DP must be within the lower clause. Thus, our initial assumption about licensing taking place under strict sisterhood must be weakened at least in the case of case assignment.²³ I propose that all licensers except for case assigners require their licensees to be sisters as before. Case assigners, though, will require only c-command. When case assignment takes place in conjunction with other (non case) licensing relations, strict sisterhood will still be required.²⁴ We should therefore view additions to the structure which we are building not as attachments in the standard sense but rather as additions to a “database” of domination, direct domination and precedence statements in style of D-theory of [Marcus *et al.*, 1983] and [Marcus and Hindle, 1988]. Each instance of licensing will augment the description of the wss by adding a new domination statement between the

²³ An important question to be answered is why this is so. Why should case assignment be subject to weaker structural requirements than the other forms of licensing?

²⁴ We might also consider adding the requirement of adjacency to case assignment as in [Stowell, 1981]. We need not add an adjacency predicate to our structural database since adjacency is implicitly represented through incrementality of input. See also the analysis of rightward extraposition and heavy NP shift for an interesting exception to this implicit representation.

parent of the licensing head and the licensee as well as a precedence statement between the licenser and licensee (depending upon the directionality, of course). Case assignment, then, differs from other licensing in that its domination statement will be just simple domination whereas other licensing relations will posit direct domination. Introducing domination into the structural representation adds a bit of complication to the evaluation of domain size and to the determination of licensing possibilities. For the evaluation of domain size, we must decide whether a node which is not directly attached (in the standard sense), is to be counted in the evaluation. The default answer is yes and so far, there do not seem to be any problems stemming from this. In so far as the determination of licensing possibilities, we must extend the domain of possible attachments to the wss.²⁵ Previously this was exactly the right frontier. I propose that a node which is not attached via direct domination will not isolate its left sister from possible licensing to the right. This will serve to enlarge the search of possible attachment sites, but since our structures will remain of bounded size, the search will be no worse in terms of complexity.

Returning now to the example at hand, we attach the DP to the V^0 . Since the only relation involved is case assignment, we posit dominance between the V' and the DP. Next, we unsubstiteute the DP since it constitutes an independent predication. We are left with the following wss²⁶:

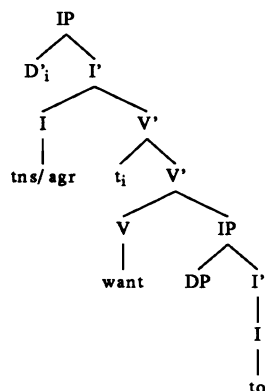


Now, we need not resolve the unsatisfied theta give on the V^0 node since the domination link does not isolate it from the right frontier. We continue with the next token, *to*, projecting it to IP. When we now consider attachment possibilities we have three. First, we have the attachment of the wss as the subject of the newly projected IP. Second, we have the attachment of the DP as subject of the new IP. Third, we have the attachment of the

²⁵Note that this extension also affects which nodes are considered to be internal nodes which need their gives resolved through some empty category.

²⁶The dotted connection signifies a dominance relation between the nodes rather than the standard direct dominance.

new IP as theta receiver of the verb's internal theta role. Since the ordering of attachment preferences dictates that theta attachments are preferred over subject attachments, we employ the third possibility. We now attach the IP as sister to V^0 . How do we deal with the dangling DP? We know that it is dominated by the V' . In addition, we now know that it is dominated by the IP node. Finally, we know that it precedes the I^0 node as a result of its position in the input string. Now, are there any attachments which are consistent with these constraints? The answer is yes. The DP may be attached as subject of the lower IP. We now have a fully determined structure - that is, there are no dominance relations which have not been expanded into a string of direct domination links.²⁷



At this point we need to perform an unadjunction, in this case we will remove the upper clause. Then we will resolve the unsatisfied theta need on the DP by pushing a copy of this node onto the trace stack. We will conclude the parse as above.

There is an interesting consequence of this “restructuring” analysis. Recall that one of the major claims of this work is that all constraints of grammar are expressible over small bounded domains, our elementary tree structures. In general, these correspond to the structures we are building as the wss. Thus, any constraint which is statable over the dynamically constructed wss could be stated over an elementary tree as well. In fact, we argued above that this was desirable since we have a time at which all constraints must be satisfied, particularly upon domain reduction. Now, though, the structure we are building upon assignment of exceptional case, which includes the dominance links, is not represented as one of the final elementary trees. Thus, we have a slightly extended domain over which constraints might be stated and satisfied. The question is: is there some principle

²⁷A research question which remains is how long we let these dominance links persist. I conjecture that they must be resolved as soon as they are not on the right frontier any longer. Note, though, that since any structure which is connected by only a dominance link is still included in our domain reduction process, any potential complexity problems will remain trivialized as a result of the bounded structures.

of grammar which requires and thereby motivates the use of such domains?

Consider the problem of binding of anaphors - the so-called condition A of the binding theory:

(40) Anaphors must be bound in their governing category.

The naive translation of this constraint into the language of the TAG framework would be:

(41) Anaphors must be bound in their elementary tree domain.

In some sense, this is a natural translation. A governing category is analogous to the licensing domain of a given element. This simple formulation works for the core cases of English anaphor binding such as²⁸

(42) Virginia_i convinced herself_i that there is a Santa Claus.

Both the subject, the DP dominating *Virginia* and the object, *herself*, are present in the same elementary tree. The interaction of raising and anaphors also seems to work correctly:

(43) Harvey_i is likely to kill himself_i.

Harvey and the anaphor *himself* will be present in the single elementary tree into which the raising predicate is adjoined. However, it quickly becomes apparent that this formulation of the binding condition is not sufficient. In cases such as

(44) Zippy_i believes [_{IP} himself_i to be a pinhead]

the anaphor is located within the lower clause - it is the subject of the predicate *be a pinhead* - and hence is associated with the elementary tree determined by the lower predication. Thus, it is not bound within the single elementary structure, yet it is well formed. Note, however, that this sentence contrasts sharply with one in which the lower clause is tensed.

(45) * Zippy_i believes [_{IP} himself_i is a pinhead]

The ungrammaticality of this example arises from the binding problem since a similar example without the anaphor is acceptable.

(46) Zippy believes [_{IP} Dan is a pinhead]

²⁸We must assume that the "element" which needs to be bound is really the root of the DP projection. The whole DP projection will be unsubstituted during domain reduction.

I claim that the relevant distinction between the two examples of anaphors lies in the fact that in the first, the anaphor is initially licensed by the higher clause through case assignment, whereas in the second, it is only licensed as the subject of the lower clause. Once the anaphor is licensed in the higher clause, it is associated with this structure in at least some temporary manner. It is over this extended domain that we will allow binding of anaphors to take place. We now restate our version of principle A.

(47) An anaphor must be bound within the WSS into which it is attached.

There are, of course, many cases of anaphors which even this condition will not cover.²⁹ However, this suggests a potential solution to seeming shortcomings to the strict locality imposed by the TAG framework.

We have now seen that the parser offers two different domains of locality. One is the grammatical domain of elementary tree structures over which semantic and certain abstract grammatical constraints are checked. The other is the domain of processing, the WSS. It is over this domain that licensing relations hold. Ordinarily the divisions which these domains maintain are quite similar. In general, licensing relations, notably theta relations, hold within the domain of an elementary tree. However, in this section we witnessed a particular construction, ECM, in which there is a sharp division between nodes which are processing domain co-occurrent and grammatical domain co-occurrent. This processing domain provided us with necessary extension to explain constraints on the binding of anaphors.

²⁹The following examples spring to mind immediately:

Anthony_i continuously re-read the book about himself_i;
Lois_i thinks that naked pictures of herself_i should not appear in print

One thing which all of these cases have in common is that the anaphors lack an “accessible subject” within what would be the WSS domain. Perhaps we can restrict the notion of elementary tree as a predication to include an explicit subject. In the first example, then, *himself* would appear in the same tree as the verbal predication since the DP lacks a subject. We could, however, rule out

* Anthony_i continuously re-read Anne’s book about himself_i;

since the object DP constitutes a predication and thus *himself* will never appear in the same WSS as the sentential subject. A major problem with this approach, though, is that we can get unboundedly large elementary structures.

Anthony_i continuously re-read a book about a recording of a speech . . . about himself_i;

This difficulty waits for future work to be resolved.

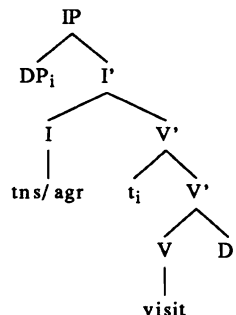
3.6.4 Modification

In this next section, let us consider how the parser deals with attachment of modification structures. Up until this point, all licensing relations that we have seen to motivate attachments have either been established reciprocally (i.e. the licenser has a give which is satisfied by the licensee which in turn has a need satisfied by the licenser) or have been established in the direction of licensing (i.e. the licensee satisfies a give of the licenser). There is a third possibility: that licensing may be established in the opposite direction, from licensee to licenser. Modification licensing is an example of this third possibility. Why must this be the case? Recall that a well formed structure has no unsatisfied gives or needs associated with any of its nodes. If an attachment of a modifying phrase is to be motivated by (at least) a give, then every node which is a potential attachment point must have a give associated with it of the appropriate type. However, all of these possibilities are surely not guaranteed to be utilized in a single sentence and therefore we will not regard the final structure as well formed. It is possible to modify the condition on well formed structures to allow for some unsaturated gives and needs, but then we would need to motivate the distinction between modification gives and the others. Moreover, there is no way, using a bounded number of gives in any projected node, to represent the fact that we may have an arbitrary number of modifications at a single node. Thus we must use needs to drive the attachment of modification structures.

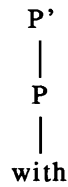
Let us examine the parser's performance on the following sentence:

(48) Arthur visited the zoo with the gorilla.

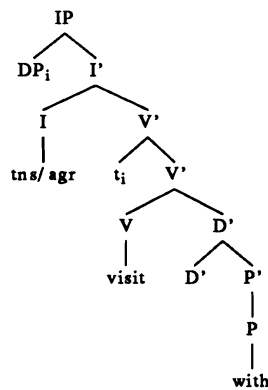
After parsing the sentence up through the prepositional phrase, we have as our WSS:



Also, the trace stack will be empty. Now, *with* is projected as



Associated with the P^0 node are a rightward theta give for an element of category DP and a rightward accusative case give. Associated with the P' node is a need of type modification. Since there are no unsatisfied gives on the right frontier of the wss, the only relevant constraint is this modification need. Thus, the possible attachment sites for the P' are entirely determined by the value of the function SN. This function determines that possible attachment sites are at the D' and V' nodes.³⁰ From our attachment preference ordering, we see that for attachments of same type, low attachment is preferred and we thus attach the P' to the D' yielding:



A couple of points about the choice of attachment site are in order. First, a word about the attachment ordering. The attachment ordering which I gave above, reflects what, I believe, are the null context preferences of the human syntactic processor. Certain relative orderings of licensing types are necessary, such as functional selection before theta assignment, so that even simple sentences may be parsed, as has been seen above. Others need to be tested using the methods of psycholinguistics. The major point I want to make with such an ordering, though, is that the different attachment types need to be distinguished and relativized. Simple tree height metrics, low attachment for example, will simply not work. Now, even as presented, the ordering is not inviolate. Pragmatic, semantic

³⁰This selection is not intended to make any particular claims about the syntax of modification. If we prefer to assume that the attachment of sentential modifiers is also possible to IP, that is fine. Crucially, though, we assume that nominal modifiers appear on DP since the internal structure of this node is unavailable having been unsubstituted during domain reduction. It may be possible to relax even this assumption if we change the control structure which decides to close the construction of the DP node.

and intonational factors will most certainly change preferences. However, I maintain that the rich information contained in the types of the licensing relations associated with each attachment possibility is necessary for a proper interface to these other linguistic modules.

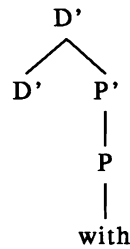
The second point about attachment choice concerns the interaction of domain reduction and attachment. Once an elementary tree is unadjoined or unsubstituted, we do not consider it to be available to the syntactic processor. The only candidates for attachment sites are the nodes present in the wss. Thus, when we are parsing a bi-clausal sentence, once we enter the lower clause, no modification is possible to the higher clause. This restriction mirrors the intuition that material from the current clause, or the current clause itself, is most easily construed as the modified object.

However, this is clearly too strong. We can certainly construct contexts in which attachment is possible to non-minimal elements.

(49) I told the woman to take a bath after I had smelled her odor.

This is a problem for the parsing model as I have proposed it. One possible solution is that attachment of modification structures (as well as any other need-driven attachments) is achieved in two ways. The first is the method we have been discussing - attachment directly in the wss. However, a second manner of attachment might be that the attachment of such phrases is done at a semantic level. It is unclear how such an attachment will interact with the rest of the parsing machinery, particularly with respect to the domain evaluation and reduction process. I speculate that such attachments may only take place with a focused entity. Let us also assume that such a focus is unique for a given sentence. Thus, we predict that attachments will only be possible clause internally or to one other entity, clausal or nominal. Whether this prediction is correct or whether the distinction between the two sorts of attachment may be maintained awaits further investigation.

Returning, now, to the example at hand, we must perform domain reduction since the prepositional projection constitutes a predication independent of the verbal predication. There is a recursive structure which we can unadjoin in this case, having the higher DP as its root and the lower DP as its foot, since all of its non-right frontier nodes are satisfied. Upon unadjunction, we note that the newly unadjoined structure does not have a satisfied right frontier. Thus we push the old wss onto the context stack along with the (empty) trace stack and continue with this modification structure as our wss.



Next, we parse the DP, *the gorilla*, by first attaching the D projection as sister to the P⁰, then unsubstituting, again pushing on the context stack. Then, we project and attach the N as sister of D⁰. We see this DP structure is satisfied and the context stack is not empty, so we pop the context stack. We again see that the wss the restored PP modification structure, is satisfied and the context stack is non-empty so we pop one more time. Now, we again have the satisfied main verbal predication as our wss and we are finished.

Recall that needs are non-directional as we have defined them. Thus, we will also be able to derive the following with the same assignment of needs:³¹

- (50) During the lecture, the student noticed the professor had no clothes on.

The modification need of the P' is satisfied either from a node to the right or to the left.

However, there do seem to be some cases in which we do want the directionality of needs, in this case modification needs, to be constrained. In English, adjectival modification is one such case: adjectives must attach to the left of the noun.

- (51) i. the outrageous taxes
 ii. *the taxes outrageous

This suggests that we need to augment our current formulation of needs to include directionality.

One might however suggest that maintaining needs without directionality is correct. Rather, it might be proposed, contrary to the previous discussion, that modification gives should be added which are possibly unsatisfied in well formed structures. Now, however, contrast the English situation with that of French in which some adjectives attach post-nominally whereas others are pre-nominal.

³¹Actually, this is not quite true. The problem, though, is independent and lies in the lack of IP node at the relevant point. More on this point later.

- (52) i. la soupe americaine
 *l'americaine soupe
 ii. *la fille jeune
 la jeune fille

This suggests that our formulation of modification as a need is correct. Why? If modification were stated as a give, French nouns would require two distinct gives, each restricted to apply only to the correct class of adjective. This would cause a great deal of complication in the computation of the function **SG** in such cases. Thus far, **SG** has only considered such primitive features as node category and bar level. The lexical or semantic class of the projection head seems beyond the range of information we would like the low-level function **SG** to consider. If, instead, we utilize the directional needs analysis, we can place the “complexity” into the acquisition of the lexical item. The directionality is determined at projection and is not computed during the consideration of each attachment possibility.

I would like to point out that once we allow directional needs, we must still allow directionality to remain unspecified so that we correctly capture the case filter and the theta criterion. These principles make no claims about the direction from which a given node receives case or theta role, but only requires some assignment.³²

3.6.5 Modifier Extraposition and Heavy NP shift

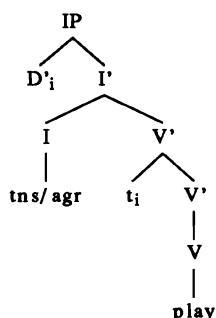
The reader may have noticed that in the treatment of modifiers we have seen above, after a modifying phrase is removed, it no longer prevents access to the right frontier of the WSS. In fact, the WSS is identical before and after the processing of modifying adjuncts. Therefore, intervening modification structures will not prevent the licensing of nodes by previously “obstructed” nodes.

This property of the parser has as a consequence an analysis of some cases of rightward “movement,” particularly heavy NP shift as well as modifier extraposition. Take an example of heavy NP shift:

- (53) The pianist played without a flaw the piece which Johann had composed for her.

Just before the receipt of *without*, the WSS will be:

³²It is of interest why there do not seem to be any directionally unspecified gives in English. This perhaps has something to do with the rigidity of English word order.



Now, we project and attach the PP, perform the unadjunction, process the DP, and pop everything until the original wss is back. We now continue by attaching the direct object DP, exactly as we would have had there not been any intervening modifier.

Similarly, the parser will allow the “extrapolation” of adjuncts across other adjuncts.

- (54) The conductor chastised the clarinetist yesterday who squeaked during the performance.

The simple transitive sentence is parsed as before. The sentential modifier *yesterday* then attaches, say, to the IP, thereby closing off the entire right frontier as possible nodes for future attachment. Next, however, an unadjunction is performed removing the adverb. Thus, the right frontier remains accessible for further processing and we can attach the relative clause modifying the object DP.

Interestingly, this account predicts that we will not be able to extrapolate across other arguments as in:

- (55) * The conductor is giving the orchestra a lecture that had screwed up the performance.

Once the direct object, *a lecture*, is attached to the IP, a DP node is created which will persist and thereby prevent access to the indirect object. The node will not be removed through adjunction since it is an argument of the verb, the predicate which determines the elementary tree. Thus, we have a predicted asymmetry between arguments and non-arguments.

Note that this account of rightward movement does not consist of movement at all. Rather, we are allowing derived, although not explicitly computed, structures which are strictly non-treelike since they contain crossing branches. Thus, our treatment of rightward movement is rather different from that of leftward movement. Constraints on movement resulting from properties of empty categories will not apply to the rightward movement cases. Whether this sharp distinction is appropriate remains for the data to determine.

If we review the nature of movement and the way in which empty categories are motivated in processing, we see that we could not have pursued a different sort of account rightward extraposition of adjuncts, particularly one which utilized real chain formation. In such a case, we would have been required to posit a trace before seeing the moved category itself. However, the notion of trace in the parser was just the re-utilization of a previously attached element after it is popped from the trace stack. Moreover, we could not have posited a non-trace empty category as the initial location of the extraposed adjuncts since they are not independently required by some unreachable give. Rather, the adjuncts are licensed as result of the satisfaction of their own modification need.

This mechanism, however, cannot account for extraposition of subject modifiers, however.

(56) A ballerina defected who had never been to America before.

The subject will remain inaccessible once it is attached as sister of I' . Another shortcoming of this analysis is that it does not explain why NPs need to be “heavy” in order to be grammatically extraposed. Contrast the above example of heavy NP shift with the following:

(57) ?? Harvey discussed during the first hour of the party Physics.

We might pursue some explanation relating to the constraint of Stowell 1981, that all case assignment takes place under adjacency. Thus, even though a case give is accessible to the right frontier, a DP might not be licensed after a modification structure is unadjoined since adjacency no longer holds. Of course, why this adjacency requirement fails to hold in the case of heavy NPs would need to be worked out.

Additionally, we have no explanation of why arbitrary alternation of modification attachment are unacceptable:

(58) ?? Joe saw the seal [yesterday] [with the white fur] [after lunch] [which had long tusks]

We might want to appeal to some processing difficulty in alternation of attachment site, but thus far this is not imposed by the machinery of the parser.

3.6.6 Genitives

Genitive DPs are handled in much the same way as tensed sentences. The $'s$ projects to two bar levels. The D' has a leftward case give and the D^0 has a rightward function select

give of type N. In parsing the DP “Ariel ’s picture,” we first project *Ariel* to DP, project ’s and attach the DP in the wss as sister to D’. The specifier is then unsubstituted as before. Now, the DP node, the unsubstitution site, is copied onto the trace stack since it has an unsatisfiable theta need. We continue with the N’ projection of picture with a rightward theta give on the N⁰ node. It is attached as sister of D⁰ from the function select give. Let us assume that we are now continuing with a tensed sentence. Thus, we receive *tns/agr* and project it to two bar levels with gives and needs assigned as before. The DP we have been building in the wss is attached as the specifier of I and is then unsubstituted. However, upon unsubstitution, we see that there is an unsatisfied give on the frontier and a node on the trace stack which will be removed. This node on the trace stack, the DP, is attached as sister to the N’, its internal argument, forming a chain with the specifier of D. All constraints within the unsubstituted DP are now satisfied. The remainder of the sentence is processed as before.

This processing is driven by the fact that the N assigns some theta role. This is somewhat believable in the case of *picture*. However, it is much less so for “non-relational” nouns such as *cake*. It would seem rather stipulative that all nouns assign a potential possessor theta role. Yet this is exactly what we need in order to allow *Lucy’s cake* to be parsed. What happens to this theta role when it is not assigned to some overt argument? Will there be a PRO internal to the projection of N in such cases? What is the nature of this theta role? It does not seem that the notion “possessor” is entirely adequate. The relation between the DP and the head noun can be quite varied for compare the following:

- (59) Lucy’s cake
 Lucy’s birthplace
 Lucy’s despair

I note this is a problem not only for the parser, but for GB theory in general. We must have some coherent explanation about how such genitives receive theta roles. Otherwise the theta criterion will be factually incorrect.

3.7 Remaining problems

In closing this chapter, I would like to point out a couple of technical problems which remain with the parser.

3.7.1 Incompleteness

The first problem with the parser is its incompleteness with respect to the system of licensing relations. That is, the parser will not be able to recover valid structures for all word strings for which there exist such structures. This contrasts with the converse property of soundness, that all structures constructed are valid structures. The parser is sound with respect to the licensing relations.

To see that the parser is incomplete, consider the following example:

(60) Sheila knows that Ted eats fish

It is clear that the parser will be able to handle the initial subsentence up through *know*. The token *that* will project to C' . This projection is attached as sister to V^0 as a result of the theta need of the verb. The only need on the C projection is a rightward function select of type I on the C^0 . Now, when *Ted* is projected to DP, there is no licensing relation which is available to cause this DP to be attached to the wss. Thus, we fail. However, it is clear that a valid structure does exist. This subject DP is licensed by the I' node, determined by the *tns/agr* features of the lower clause, via case assignment. This IP may then attach as sister of the C^0 satisfying the function select.

The problem seems to be that a continuous chain of licensing relations will not necessarily exist in a single left to right pass through the input. English is rather surprising with respect to how many structures are parsable using such a simple method. Had the initial investigation been focused on head final languages, we would certainly have reached this impasse much earlier. Nonetheless, it is clear that we must allow for some method of postponing the necessity of attachment of a projected lexical item. There are a wide variety of possible mechanisms for accomplishing this task, but they are not considered here since these concerns seem largely orthogonal to the framework being developed here.³³

A similar problem crops up in the case of object DPs having genitive subjects. These genitives will be attached as object to the verb since the parser employs a simplistic greedy attachment strategy. It does not break at this point as was the case with subjects of embedded clauses because there is no overt morphology distinguishing a genitive DP and

³³One possibility consists of allowing for context stack pushing of the WSS and continuing with attempts to process this unattachable projection as the WSS. After this projection is licensed by some other later projection, we attempt to reintegrate this structure back into the original WSS. This seems reminiscent of the attention shifting mechanism of Marcus 1980. Of course, we need some method to constrain the use of this device.

thereby preventing it from being licensed through accusative case assignment. [Abney, 1986] proposes a solution to this problem by allowing the “real object” to *steal* the prematurely attached object without interrupting the flow of processing. Such local adjustments are not equivalent, in his model, to real necessitated backtracking and garden pathing. In languages where case is morphologically represented, though, we will experience the same problem that we had with the embedded subjects since they will not be allowed to be attached as object at all.

3.7.2 Wh-movement and Phonologically Empty Heads

Given that TAG provides an elegant way of locally specifying constraints on long-distance movement, it is unfortunate that Wh-movement has been absent from the discussion of linguistic phenomena which are handled by the parser. The problem directly stems from the GB theory explanations of the cause of Wh-movement and the licensing of the Wh-element once it is in its landing site. Fukui and Speas [1986] propose that there is a phonologically empty complementizer which has f-features that license the appearance of wh-elements in the specifier of C. However, they do not give an answer to the problem of why the wh-element must move at all. Thus, there does not seem to be any reason for supposing that vacuous movement has taken place in

(61) Who kissed the milkman?

It is consistent with everything we have said thus far that *who* remains in the specifier of I position. There can clearly be no need associated with the DP projection of a Wh-element which causes it to attach as sister to some appropriate complementizer. Otherwise, we would be unable to generate echo questions such as:

(62) You ate what?

in which the wh-element remains in its VP internal position.³⁴

With respect to object extraction, though, we must posit some form of movement. However, in relative clauses, no overt complementizer appears.

(63) the tape [_{CP} which_i Rose erased *t_i*]

³⁴It might be possible to say that the Wh either needs proper (lexical) government OR it needs the appropriate wh features. In that way, we could account for superiority effects. Also, we could provide some analysis of why wh-movement takes place in terms of, albeit disjunctive, needs.

If we suppose that the final position of the *wh*-element is in specifier of *C*, we must posit the existence of some empty head. For typical *wh*-questions, if we consider the *do* in English to be the phonological counterpart of the *wh*-licensing complementizer, then the analysis will follow as soon as we solve the problem of incompleteness

At present, I do not have any theory of positing empty heads. It is likely that this will interact with the solution to the incompleteness problem. We must develop some strategies for deciding when it is appropriate to attention shift and when we must suppose that some phonologically non-overt material needs to be posited.

3.7.3 Head movement

In the parser, the only type of movement we have considered is that of maximal projections. However, GB allows another type of movement, so-called head movement. In such cases, a head is moved, typically adjoined to another X^0 node. This is often thought to be the way in which verbs acquire their tense, either through raising of V^0 to I^0 or lowering I^0 to V^0 . Thus far, we have abstracted away from this process by assuming that the verbal and inflectional heads remain distinct at *S*-structure via some morphological pre-processing. However, this abstraction cannot always be maintained. Morphological pre-processing will not undo the sorts of head movement which are proposed for analyses for V2 in Germanic.

Unfortunately, in the current framework, it is unclear how we might allow for moved heads. As soon as a lexical item is encountered in the input string, it is projected to its maximal projection. There is no option to attach just the head, and leave the remainder of the projection for later to be “projected” in the absence of any lexical information. This problem’s solution will undoubtedly interact with the general problem of positing empty heads, similar to our current relationship between positing traces and other empty categories. This remains for future work.

Chapter 4

Conclusion

In this thesis, I have presented a model for the use of knowledge of language as represented by GB theory. As I argued in chapter 1, it is desirable to maintain a maximally direct mapping between the grammar along with its set of representations and the processing mechanism, while still maintaining nice computational properties.

How well has this model succeeded in satisfying our desiderata? The use of generalized licensing allows for the natural and direct expression of a substantial class of grammatical constraints. The addition of TAG to maintain a small bounded working space has allowed us to maintain linear time performance. The combination of these two seems to have accomplished just what we were seeking.

One might, however, argue that we have backed off much further from a direct instantiation of the theory than was necessary. In particular, the analysis of movement in this system is quite a bit different from the freely applied move- α rule of GB theory. Essentially, we only allow very particular types of movements to be considered at all. However, this is not necessarily a fatal blow to our enterprise. In fact, it seems rather likely that the only movements that are ever really utilized belong to the class we consider or some natural extension thereof. Thus, the superficial generality of this portion of GB theory may be just that.

This is not to say that we have been completely victorious. Throughout chapter 3, I mentioned shortcomings of the mechanism I have proposed. The incompleteness of the parser is quite substantial. That is, there is a large class of examples which the mechanism will not handle. Of course, it is not necessarily desirable for the parser to be able handle all possible inputs for which there exists a well-formed set of satisfied licensing relations.

Garden path and center embedded sentences are two cases where we might not want our mechanism to succeed. However, we should be able to parse such seemingly simple constructions as sentences with tensed embedded clauses as well as Wh-questions. On top of this and other technical problems, though, remain the issues of ambiguity and optionality of arguments. For the course of this work, I have completely ignored them. These are major hurdles which must be crossed before this is to serve as a real model for language processing.

All of these problems, though, can be characterized as problems with the mechanism that has been proposed. Subsequent work will certainly have to reshape the computations performed by the parser. In contrast, I think, is the representation I have proposed, utilizing generalized licensing coupled with TAG. Here, I think a stable formulation has been reached which has already begun to provide us with insights into the shape of the meta-grammar which could be used by linguists to limit the class of possible grammatical constraints: that all constraints must be storable over the structural domain of an elementary tree, that is, a single predication, or it may be stated over the processing domain as was done in the case of anaphor binding.

Where do we go from here? Staying within this framework, we might see how this mechanism behaves on other languages. This will certainly require creating a more adequate set of parameters. Currently, the only parameters specified for a language are those relating to directionality. Additionally, the problem of relative freedom of word order will need to be addressed.

Another idea might be to continue on in the methodology proposed at the outset: keep exploring the space of grammar-parser relations. In order to do this in a methodical fashion will require a formalization of the intuition of “abstraction from the grammar.” If we can precisely specify the nature of the partial ordering on the objects in the space, we might be able to give a complexity theoretic characterization of the different levels of abstraction and thereby determine the appropriate level for psychological modeling of the language processor.

One final possibility for future work concerns the generalization of the idea of constraint localization that we have so crucially employed here. We might define a constraint logic of the kind that has been recently proposed in work on feature structures [Rounds and Kasper, 1986]. Such a logic would be restricted in that it could only state constraints over structures of bounded size. These structures could then be combined through recursive

constraint transformations in the style of [Pollard, 1989] to allow distant objects to be mutually constrained. Different classes of constraint transformations and primitive object types could be investigated for expressive adequacy in specifying grammatical principles.

Bibliography

- [Abney, 1986] Steven Abney. Licensing and parsing. In *Proceedings of NELS 16*, Amherst, MA.
- [Chomsky, 1981] Noam Chomsky. *Lectures on Government and Binding*. Foris, Dordrecht.
- [Chomsky, 1986a] Noam Chomsky. *Barriers*. MIT Press, Cambridge, MA.
- [Chomsky, 1986b] Noam Chomsky. *Knowledge of Language, Its Nature, Origin, and Use*. Praeger, New York, NY.
- [Chomsky, 1989] Noam Chomsky. Some notes on economy of representation and derivation. In *MIT Working Papers in Linguistics 10*, MIT Department of Linguistics.
- [Dorr, 1987] Bonnie Dorr. *UNITRAN: A Principle-Based Approach to Machine Translation*. Technical Report 1000, MIT Artificial Intelligence Laboratory.
- [Fodor, 1978] Janet D Fodor. Parsing strategies and constraints on transformations. *Linguistic Inquiry*, 9.
- [Frank, 1988] Robert Frank. Projection, predication and tree adjoining grammar. draft, University of Pennsylvania.
- [Frank, 1989] Robert Frank. A note on raising in combinatory grammar. manuscript, University of Pennsylvania.
- [Fukui and Speas, 1986] Naoki Fukui and Margaret Speas. Specifiers and projection. In Naoki Fukui, T. Rappaport, and E. Sagey, editors, *MIT Working Papers in Linguistics 8*, MIT Department of Linguistics.
- [Heycock, 1989] Caroline Heycock. On the subject. manuscript, University of Pennsylvania.

- [Higginbotham, 1984] James Higginbotham. English is not a context-free language. *Linguistic Inquiry*, 15.
- [Higginbotham, 1985] James Higginbotham. On semantics. *Linguistic Inquiry*, 16.
- [Huang, 1982] James Huang. *Logical Relations in Chinese and the Theory of Grammar*. PhD thesis, MIT.
- [Johnson, 1988] Mark Johnson. Parsing as deduction: the use of knowledge of language. In *The MIT Parsing Volume, 1987-88*, MIT Center for Cognitive Science.
- [Joshi, 1985] Aravind Joshi. How much context-sensitivity is required to provide reasonable structural descriptions: tree adjoining grammars. In D. Dowty, L. Karttunen, and A. Zwicky, editors, *Natural Language Processing: Psycholinguistic, Computational and Theoretical Perspectives*, Cambridge University Press.
- [Joshi *et al.*, 1975] A. Joshi, L. Levy, and M. Takahashi. Tree adjunct grammars. *Journal of the Computer and System Sciences*, 10.
- [Kashket, 1987] Michael Kashket. *A Government-Binding Based Parser for Warlpiri, a Free-Word Language*. Technical Report 993, MIT Artificial Intelligence Laboratory.
- [Kayne, 1981] Richard Kayne. ECP extensions. *Linguistic Inquiry*, 12.
- [Kroch, 1986] Anthony Kroch. Unbounded dependencies and subadjacency in a tree adjoining grammar. In A. Manaster-Ramer, editor, *The Mathematics of Language*, John Benjamins.
- [Kroch, 1989] Anthony Kroch. Asymmetries in long distance extraction in a tree adjoining grammar. In Mark Baltin and Anthony Kroch, editors, *Alternative Conceptions of Phrase Structure*, University of Chicago Press.
- [Kroch and Joshi, 1985] Anthony Kroch and Aravind Joshi. *The Linguistic Relevance of Tree Adjoining Grammar*. Technical Report MS-CS-85-16, University of Pennsylvania Department of Computer and Information Sciences. To appear in *Linguistics and Philosophy*.
- [Kroch and Joshi, 1986] Anthony Kroch and Aravind Joshi. Analyzing extraposition in a tree adjoining grammar. In G. Huck and A. Ojeda, editors, *Syntax and Semantics: Discontinuous Constituents*, Academic Press.

- [Kroch and Santorini, 1987] Anthony Kroch and Beatrice Santorini. The derived constituent structure of the west germanic verb raising construction. In R. Freidin, editor, *Proceedings of the Princeton Conference on Comparative Grammar*, MIT Press, Cambridge, MA.
- [Lasnik and Saito, 1984] Howard Lasnik and Mamoru Saito. On the nature of proper government. *Linguistic Inquiry*, 15.
- [Lasnik and Uriagereka, 1988] Howard Lasnik and Juan Uriagereka. *A Course in GB Syntax: Lectures on Binding and Empty Categories*. MIT Press, Cambridge, MA.
- [Marcus, 1980] Mitchell Marcus. *A Theory of Syntactic Recognition for Natural Language*. MIT Press, Cambridge, MA.
- [Marcus and Hindle, 1988] Mitchell Marcus and Donald Hindle. Description theory and Japanese particles. manuscript, University of Pennsylvania and Bell Laboratories.
- [Marcus *et al.*, 1983] Mitchell Marcus, Donald Hindle, and Margaret Fleck. D-theory: talking about talking about trees. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, Cambridge, MA.
- [Pesetsky, 1982] David Pesetsky. *Paths and Categories*. PhD thesis, MIT.
- [Pollard, 1989] Carl Pollard. Extensions to feature structure based linguistic formalisms. Talk given at University of Pennsylvania.
- [Rothstein, 1983] Susan Rothstein. *The Syntactic Forms of Predication*. PhD thesis, MIT. Distributed by the Indiana University Linguistics Club.
- [Rounds and Kasper, 1986] William Rounds and Robert Kasper. A complete logical calculus for record structures representing linguistic information. In *IEEE Symposium on Logic and Computer Science*.
- [Schabes *et al.*, 1988] Yves Schabes, Anne Abeillé, and Aravind K. Joshi. Parsing strategies with ‘lexicalized’ grammars: application to tree adjoining grammars. In *COLING Proceedings*, Budapest.
- [Shieber, 1983] Stuart Shieber. Sentence disambiguation by a shift-reduce parsing technique. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, Cambridge, MA.

- [Shieber, 1984] Stuart Shieber. *Evidence against the Context-Freeness of Natural Language*. Technical Report 330, SRI International.
- [Stabler, 1990] Edward Stabler. Implementing government binding theories. In Levine and Davis, editors, *Formal Linguistics: Theory and Implementation*. forthcoming.
- [Steedman, 1987] Mark Steedman. Combinatory grammars and parasitic gaps. *Natural Language and Linguistic Theory*, 5.
- [Stowell, 1981] Timothy Stowell. *Origins of Phrase Structure*. PhD thesis, MIT.
- [Subrahmanyam, 1988] Ramesh Subrahmanyam. Semantic interpretation in tree adjoining grammar. In *ESCOL Proceedings*, Philadelphia, PA.
- [van Riemsdijk and Williams, 1986] Henk van Riemsdijk and Edwin Williams. *Introduction to the Theory of Grammar*. MIT Press, Cambridge, MA.
- [Vijay-Shanker, 1987] K. Vijay-Shanker. *A Study of Tree Adjoining Grammars*. PhD thesis, University of Pennsylvania.
- [Weir and Joshi, 1988] David Weir and Aravind Joshi. Combinatory categorial grammar: generative power and relationship to linear context-free rewriting systems. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, NY.
- [Weir *et al.*, 1986] David Weir, K. Vijay-Shanker, and Aravind Joshi. The relationship between tree adjoining grammars and head grammars. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, New York, NY.
- [Williams, 1980] Edwin Williams. Predication. *Linguistic Inquiry*, 11.
- [Williams, 1987] Edwin Williams. A reassignment of the functions of LF. *Linguistic Inquiry*, 18.