A Descriptive Approach to Language-Theoretic Complexity

James Rogers

September 12, 1995

Contents

1

Language Complexity in Generative Grammar

Early formal theories of syntax were generally couched in terms of rewriting systems—phrase structure grammars and the transformational grammars based on them. This approach was quite successful in uncovering structural properties of natural languages and, moreover, was theoretically very fruitful as well, yielding the field of Formal Language Theory and leading to the identification of hierarchies of language complexity classes. There was an expectation, in this early work, that such classes would play a significant role in defining the structure of natural languages. The initial definition of the Chomsky Hierarchy (Chomsky 1959) was motivated, in part, by the idea that the hierarchy might serve to characterize the class of natural languages, at least in the broad sense that some level might be shown to include the natural languages while excluding significant categories of non-natural languages. The intent was that by capturing the class of natural languages with a mechanism that had a formally well-defined generative capacity one might gain insight into the structural regularities of those languages. Further, given the dual structural/automata-theoretic characterizations of these classes one might hope not just to identify the regularities of natural language, but to account for them.

Over time the emphasis has turned away from formalisms with restricted generative capacity in favor of those that support more natural expression of the relevant regularities. These more recent approaches tend to fall into the category of *constraint-based* formalisms—formalisms that define languages by specifying constraints on the structures analyzing their strings. Concomitantly, the topic of language-theoretic complexity has all but disappeared from linguistic research. This is largely the result of a realization that the structural properties characterizing natural languages as a class may well not be those that can be distinguished by existing language complexity classes, but it is also at least in part a result of the fact that it is not at all clear how to establish such results for constraint-based theories. These studies address both of these issues. We introduce a method for establishing language-theoretic complexity results that is natural for application to constraint-based formalisms over trees. What's more, our experience in applying it to Government and Binding Theory (GB) suggests that the connection between such results and current research in natural language syntax may be stronger than generally assumed.

It will be useful, by way of introduction, to review briefly the intertwined histories of formal language theory and generative linguistics and in particular to sketch the diminishing role of language complexity within formal theories of syntax. Our focus is the tradition that has led to the development of Government and Binding Theory, but a similar transition can be found in the evolution of GPSG into HPSG, albeit accompanied, in that case, by a move from trees to a more general class of structures.

1.1 From Rewriting Systems to Constraint-Based Formalisms

In one of the earliest applications of formal language theory to natural language Chomsky (1959, 1957) undertook to prove that English is not included in the **regular languages**, and consequently, that finite-state automata are inadequate to model the human language faculty. At the same time he argued informally that the **context-free** grammars were also, if not inadequate, then at least inexpedient, for defining natural languages. This proved to be a much more difficult result to achieve and is still controversial. Although a considerable amount of subsequent work was directed towards showing formally that various natural languages were not context-free languages, that work was mostly unsuccessful (Pullum and Gazdar 1982, Pullum 1984). It is only relatively recently that compelling evidence has been offered for the non-context-freeness of natural languages, specifically based on case-marking in cross-serial verb constructions in Swiss German (Shieber 1985), on long-distance extractions in Swedish (Miller 1991), and on word formation in Bambara (Culy 1985).

While the context-free grammars are evidently too weak, the **context-sensitive** grammars seem clearly too powerful to characterize the class of natural languages in any useful way. Savitch (1987), for instance, points out that, for any **recursively enumerable** language L, there is a context-sensitive language L' such that, a string is in L if and only if it is one of an easily identifiable class of prefixes of the strings in L'.[1] Thus, the context-sensitive languages exhibit every kind of structural regularity exhibited by the r.e. languages.[2]

Natural languages, as a class, then, seem to fall between the context-free and context-sensitive languages in the sense that they include non-context-free languages but are expected to form a much smaller class than the context-sensitive languages. Even in the early work (Chomsky 1957, Chomsky 1959) there was a presumption that phrase structure grammar alone was an inadequate foundation for a theory of natural language syntax. This was based largely on the observation that a great deal of the regularity in this syntax can be accounted for by transformations that rearrange the components of sentences. At the same time, one could hope for a formal result that the class of languages generated by CFGs augmented by some transformation mechanism might be strictly smaller than the class of context-sensitive languages.

This was the intuition behind **transformational grammars** (as developed in *Aspects*, Chomsky 1965). In TG, a base grammar is associated with a set of formal transformations. Here again, the generative capacity of the grammars are well-defined, a function of both the complexity of the base grammar and the permissible transformations.[3] In one extreme, the **Universal Base Hypothesis**, the base grammar is fixed and all variation between natural languages (modulo the lexicon) is to be accounted for by variation in the transformations. The hope that such a grammar might provide meaningful bounds on the complexity of natural languages was frustrated, though, when Peters and Ritchie (1973, 1971, see also Bach and Marsh 1987) showed that transformational grammars of the type in *Aspects* are capable of generating every r.e. language. The significance of these results is not that transformational grammars are too strong in that they can generate languages that are too hard, in some sense, to be natural languages, but rather that, by itself, the hypothesis that natural languages are characterized by *Aspects*-style TGs, or even

[1] Specifically, $w \in L$ iff there is some $n \in N$ for which $\langle w, \$^n \rangle \in L'$, for all w and some symbol \$ not otherwise occurring in the language. This is the case simply because any Turing Machine that accepts L, will accept each $w \in L$ using finitely many tape cells. L' simply encodes in $\$^n$ that bound for some acceptor of L.

[2] Neither the context-sensitive nor unrestricted grammars have the same inherent connection to phrase structure that the context-free grammars exhibit through their derivation trees. Nonetheless, we assume that we are interested in the structure of language and not just the set of strings in the language. That is to say, we are interested in sets of *phrase markers* of some sort rather than sets of strings.

[3] Although one does not necessarily have the simple connection to classes of computing mechanisms exhibited by the phrase structure grammars.

by TGs with a fixed base, has no non-trivial consequences with respect to the class of natural languages. Every reasonable language is in the class of languages generated by TGs.

Thus, the idea that one could get meaningful bounds on the class of natural languages by restricting the framework in which the theory of those languages is built was largely unsuccessful. The alternative approach is to work in a framework with relatively unrestricted (or just unknown) generative capacity and characterize the natural languages by a specific theory couched in that framework. The **Principles and Parameters** approach of Government and Binding Theory, for instance, follows a strategy characterized by Berwick as aiming to "discover the properties of natural languages first, and then characterize them formally" (Berwick 1984, pg. 190). In this approach the structure of natural languages are consequences of a set of general principles that are invariant across all of the languages, and a finite set of parameters that account for the observed variation between them. While it is still usually modeled as a transformational grammar, the base grammar (*X-bar theory*) generates a single extremely simple set of structures and the catalog of transformations of *Aspects* has been replaced with a single movement rule, *move-α*—move anything anywhere. This underlying mechanism is constrained by additional principles that specify structural conditions that must be met in the admissible phrase markers. Ideally, every node generated by the base grammar and every transformation introduced by move-α is required by the consequences of these principles.[4]

GB, then, has adopted, in the place of the general formalism of TG, a specific set of instances of that formalism. It no longer suggests that TG might characterize the natural languages, rather that they are characterized by a specific set of structural principles. This leaves the question of the formal properties of the class open. But, even though the formal properties no longer have a central role in guiding the theory, it is still useful to determine these properties—both as a way of relating the GB account of language to other linguistic theories and for what these prop-

[4]While, at first glance, this might appear to be a more powerful system than the earlier TG, particularly given the relative freedom in defining the principles on which it is based, move-α, although relatively unconstrained in terms of what it can move and where it can move it, represents only a single kind of transformation. In particular, subtrees are moved from one position in the tree to some other position in the tree, leaving a *trace*—a node that is phonetically empty—behind. Thus, move-α never deletes any portion of the tree nor creates new portions. The ability of the TG-style grammars to delete material was instrumental to the proof of Peters, Jr. and Ritchie 1973. Berwick (1984) and Berwick and Weinberg (1984) argue, partly on the basis of this lack of deletion, that GB generates only recursive or context-sensitive (respectively) languages.

erties can say about the nature of the human language faculty. Unfortunately, formal complexity results are quite difficult to establish for GB. If they are to be consequences of a set of structural principles, then one needs either a complete set of those principles or precise formal bounds on the principles that can be employed. The theory provides neither of these. As a result, it is difficult to show even that the languages GB defines are recursive.

1.2 A Descriptive Approach to Language Complexity

The topic of this book is a flexible and quite powerful approach to establishing language-theoretic complexity results for linguistic theories that, like GB theories, are expressed as systems of constraints on trees. The book falls naturally into two parts—Part I introduces our approach and develops techniques for applying it and Part II gives a fully worked-out example of its application to a specific GB theory. In Chapter 3 we introduce a logical language $L^2_{K,P}$ capable of expressing many of the constraints on trees that are employed in linguistic theories. This is a monadic second-order language, allowing quantification both over individual nodes in trees and over arbitrary sets of those nodes, and is thus superficially quite expressive. In Chapters 4 and 5, however, we establish that the descriptive power of this language, in terms of strong generative capacity, is quite limited: sets of finite trees are definable in $L^2_{K,P}$ iff they are strongly context-free. Thus, any set of constraints we can capture in $L^2_{K,P}$ licenses a context-free language. Similarly, we can establish that a set of constraints is capable of defining non-context free languages by showing that they are not definable in $L^2_{K,P}$. We explore techniques for establishing such results and give examples of both kinds.

In Part II we apply this approach to Government and Binding Theory. We get both definability and non-definability results. We show, first, that free-indexation, the mechanism that is usually employed to express co-reference and agreement relationships in GB, is not definable in $L^2_{K,P}$, and thus, not enforceable by CFGs. In doing this, though, we actually get the stronger result that free-indexation, even in an extremely weak form, is capable of defining languages for which emptiness is undecidable. Thus, in general, it may not be possible to determine the consistency of linguistic theories in which free-indexation is assumed. Despite this inability to capture free-indexation, we go on to show that a set of GB principles capable of describing substantially all of common English syntax (or, rather, substantially all that has been accounted for in GB) is, in fact, definable in $L^2_{K,P}$. Thus, we are able to establish that the language licensed by a particular theory within the GB framework is strongly context-free. This gives an indication of the strength of this technique for establishing language-theoretic complexity results, as it is easily the strongest such result for a realistic GB theory that has been obtained to date.

One of the strengths we claim for this approach to language-theoretic complexity is the naturalness of $L^2_{K,P}$ as a language for formalizing linguistically interesting constraints on trees. We have tried to maintain a close connection to the linguistic concerns driving the theory throughout our formalization, and, to that end, have tried to make it as orthodox as possible. The benefit of such an approach is that, beyond its language complexity consequences, this work stands on its own as a formalization of a GB theory. The value of such formalizations, beyond providing a basis for reasoning formally about the consequences of a theory, is that they frequently raise linguistically significant issues that are obscured in less rigorous expositions. The role of free-indexation, for instance, has been questioned in a number of places within the linguistics literature (we cite some in Section 9.3). Our results provide an independent justification for such questions—the use of free-indexation in formal theories of language may be inappropriate, at least if one wants to restrict oneself to formally decidable theories. The fact that we can capture most aspects of GB without free-indexation, on the other hand, suggests that its use in such theories is unlikely to be necessary. More concretely, formalized principles may, in some cases, be simpler than the original statements of some of those principles. The identification component of the Rizzi's ECP, for example, reduces, in our treatment, to a simple requirement that every node occur in a well-formed chain. Although such results are typically only theoretically motivated, they may well suggest refinements to the original theories that can be justified empirically.

1.3 Language-Theoretic Complexity Reconsidered

Having sketched the declining role of language-theoretic complexity in the realm of generative grammar and raised the prospect of its restoration, we are left with the question of why such a restoration might be desirable.

The nature of language-theoretic complexity hierarchies is to classify languages on the basis of their structural properties. The languages in a class, for instance, will typically exhibit certain closure properties: if a language includes strings of a particular form then it includes all strings of a related form. Pumping lemmas are examples of such properties. Similarly these classes typically admit normal forms: the languages in the class can be generated from a set of simple languages using a small set of operations. Such normal forms are the topic of representation theorems such as Kleene's Theorem for regular languages or the Chomsky-Schützenberger Theorem for context-free languages (see, for instance, Hopcroft and Ullman 1979).

While the linguistic relevance of individual results of this sort is debatable, the underlying form of the results at least loosely parallels familiar linguistic issues. The closure properties of a class of languages state regularities that are exhibited by those languages; normal forms express generalizations about their structure. So while these are, perhaps, not the right results, they, at least, are not entirely the wrong kind of results. Its reasonable, then, to ask where the natural languages as a class fall with respect to these hierarchies, and, in fact, because the classes are defined in terms of their structural properties and the structural properties of human languages can be studied directly, there is a reasonable expectation of finding empirical evidence falsifying a given hypothesis about the language complexity of natural languages should such evidence exist. Note that even seemingly artificial diagnostics (like the copy language $\{ww \mid w \in \{a,b\}^*\}$) can provide the basis for such results, as witnessed by Shieber's argument for the non-context-freeness of Siwss-German (1985). On the other hand, we will show that the class of languages which can be formalized in the way we develop here can be characterized by the fact that it is possible to account for movement in these languages while respecting a fixed bound on the number of chains that overlap at any point in the tree. Effectively, this separates GB theories that license context-free languages from those that do not. We have, then, a means of characterizing context-free languages that is quite natural in the realm of GB. Thus it may well be the case that the apparent mismatch between formal language theory and natural languages has more to do with the unnaturalness of the traditional diagnostics than the appropriateness of the underlying structural properties.

By themselves these results would have little more than formal significance, but language complexity classes have automata-theoretic characterizations as well. These determine, along certain dimensions, the *types* of resources that are required to process the languages in a class. Regular languages, for instance, are characterized by acceptance by finite state automata; they can be recognized using an amount of memory that is independent of the length of the input string. Context sensitive languages can be characterized by acceptance by linear bounded automata; they can be recognized using an amount of memory that is bounded by a linear function of the length of the input. The context-free languages are probably best characterized by acceptance of their derivation trees by finite state tree automata (see Chapter 4); this corresponds to recognition by a collection of processes, each with a fixed amount of memory, where the number of processes is linear in the length of the input string and there is minimal communication between the processes in the sense that all communication with a process takes place as that process is spawned.[5] The dual characterization of language complexity classes means that hypotheses about the complexity of natural languages entail specific predictions about both the structure of those languages and the nature of the human language faculty. The key point is that these are predictions about the mechanisms implementing a particular aspect of human cognition—the human language faculty—that can be tested directly on the basis of observable behavior—the structural properties of human languages.

The possibility that such results might be obtainable is suggested by the fact that we find numerous cases in these studies in which the issues that we encounter for definability reasons, and ultimately for complexity reasons, have parallels that arise in the GB literature where they are motivated by more purely linguistic concerns. This suggests that the regularities of human languages that are the focus of the linguistic studies are, perhaps, reflections of properties of the human language faculty that can be characterized, at least to some extent, by language-theoretic complexity classes.

[5] The more common characterization of the context-free languages by acceptance by push-down automata seems too closely tied to an idiosyncratic model of computation to correspond to a natural restriction on the types of resources.

Part I

The Descriptive Complexity of Strongly Context-Free Languages

2

Introduction to Part I

The first half of this book is an exploration of the theory of variably branching finite trees in a logical language that allows formal reasoning in terms of the kinds of relationships between nodes in a tree that generally form the foundation of formal theories of syntactic structure: in particular the relationship between a node and its children, the relationship between a node and the nodes in the sub-tree it dominates, and the relationship between a node and the nodes that precede it in the left-to-right ordering of the tree. The language supports reasoning about labels or sets of features attributed to nodes through monadic second-order variables—variables that range over arbitrary sets of nodes. Any bounded system of attributes can be interpreted as variables to which are assigned the sets of nodes exhibiting those attributes. As a result, most systems of constraints on trees can be expressed nearly immediately in the language. Exceptions include systems that, like HPSG, potentially distinguish infinitely many types of nodes.[1]

The key motivation for formalizing linguistic theories within this logic, as opposed to the variety of other formal systems that are available, is the primary result of this part—languages are definable in this logic iff they are strongly context-free. Thus, in addition to the benefits accrued from any rigorous formalization of a theory, one gets strong generative-capacity results for the language in question. This result, of course, implies that there are limits to the range of constraints that can be expressed in this language. We explore the nature of the constraints that cannot be captured directly and those that cannot be captured at all, and we provide examples of techniques both for defining constraints and for proving non-definability of constraints.

The content of this part is necessarily quite technical and it pre-

[1] But, based on the analyses of the second half of this book, one might question the necessity of such a large variety of categories.

sumes some familiarity with standard concepts of mathematical logic. We provide definitions for most of the concepts we employ, however, and it should be accessible to most mathematically inclined readers. Those who are primarily interested in applying these results can safely skip most of the details, focusing on Chapter 3 and Sections 5.1, 5.2, and 5.4, and skipping, perhaps, most of the proofs. Part II presents a detailed formalization of a quite large theory of English using this approach. In doing so, it demonstrates a variety of techniques for capturing a wide range of constraints and for circumventing the superficial limitations of the language. Thus, supplemented with only some of the background from this part, Part II should serve as an in depth introduction to the application of these results to linguistic theories.

3

Trees as Elementary Structures

There have been two dominant approaches to the formalization of trees. One of these, an algebraic approach (see, for instance, Courcelle 1983), has grown primarily from studies in the semantics of programming languages and program schemes. In this approach, trees interpret terms in the algebra generated by some finite set of function symbols. The term $f(x,y)$, for instance, is interpreted as a tree consisting of a root labeled f that has the subtrees x and y as children. Maher (1988) has provided a complete axiomatization for the equational theory of these trees. For our purposes there are two characteristics of this theory that are most significant: in it one reasons about (variables range over) entire trees as opposed to individual nodes in those trees, and it is extensional in the sense that $f(x,y) = f(g(a), g(a))$ implies that $x = y$.

In contrast, the second approach is concerned with the the internal structure of trees. Formal treatments of trees of this sort are ultimately founded in the theory of multiple successor functions, a generalization of the theory of the natural numbers with successor and less-than. The domain of this theory is the individual nodes in the tree—one reasons about the relationships between these nodes. Here, it is a theorem that the left successor of a node is not equal to the right successor of that node regardless of how the nodes are labeled. The structure of multiple successor functions is an infinite tree in which all nodes have the same (possibly infinite) degree. Its language includes predicate symbols for each successor function, a predicate symbol for domination, and one for lexicographic order (the total order imposed by domination and the ordering among the successor functions). Rabin (1969) has shown that SnS, the monadic second-order theory of this structure with n successor functions, is decidable for all $n \leq \omega$. An axiomatization of the weak monadic second-order fragment has been provided by Siefkes (1978).

In applications to computational linguistics trees typically represent the relationships between the components of sentences. Here, it is the second approach that is appropriate. One wants to distinguish, for instance, between identical noun phrases occurring at different positions in a sentence. The relations of interest are based on the relation of a node to its immediate successors (**parent** or **immediate domination**), the relation of a node to the nodes in the subtree rooted at that node (**domination**), and the left to right ordering of the branches in the tree (**linear precedence** or **left-of**). Here it is often useful to be able to reason about domination independent of parent. Such reasoning is supported directly by SnS. On the other hand, it is also often useful to reason about the parent relation independent of left-of. Since left-of is derived from the ordering of the successor functions in SnS, it is inconvenient in this respect. Further, these applications are concerned with (typically) finite trees with variable branching, in contrast to SnS, which is the theory of an infinite tree with fixed branching.

In this chapter we provide a an axiomatization of variably branching trees in a signature tailored to linguistic applications. While our structures differ from the structure of SnS (in that they vary in size and branching) and our signature varies (in that it is relational and expresses left-of independently of parent) we will show, in Chapter 4 that the theory of this class of structures can be expressed as a fragment of SωS. Consequently, the theory is decidable, even though it is not complete.[1]

3.1 Language

The signature we employ is intended to support expression of properties of trees that typically occur in linguistic theories in as direct a manner as possible. It includes predicates for the standard tree relations: parent, domination, **proper domination** (irreflexive), left-of, and equality. It also includes individual constant symbols (constants naming specific nodes in the tree) and monadic predicate symbols (constants naming specific sets of nodes) as may be required. These latter can be thought of as atomic labels—the formula NP(x) will be true for every x labeled NP. There are two sorts of variables: those ranging over individuals and those ranging over sets. Thus this is a **monadic second-order** language. Crucially, this is all the language includes. By restricting ourselves to this language we commit ourselves to working only with properties that can be expressed in terms of these basic predicates.

[1] It is incomplete (does not include either ϕ or $\neg\phi$ for some formulae) because we include trees of all countable cardinalities. A common approach to establishing decidability of a theory is to show that it is recursively axiomatizable and complete. The theory of trees is recursively axiomatizable (even in the first-order language—see Backofen et al. 1995), but, as it is incomplete, this approach is not applicable.

To be precise, the language depends on the sets of individual and set constants it employs. In general, then, we think in terms of a family of languages parameterized by those sets.

Definition 1 For $\boldsymbol{K}$ a set of individual constant symbols, and $\boldsymbol{P}$ a set of predicate symbols, both countable, let $L^2_{K,P}$ be the language built up from $\boldsymbol{K}$, $\boldsymbol{P}$, a fixed countably infinite set of variables, in two ranks, $\boldsymbol{X} = \boldsymbol{X}^0 \cup \boldsymbol{X}^1$ and the symbols:

$\triangleleft, \triangleleft^*, \triangleleft^+, \prec$ — two place predicates, *parent*, *domination*, *proper domination* and *left-of* respectively,

$\approx$ — equality predicate,

$\wedge, \vee, \neg, \forall, \exists, (,), [,]$ — usual logical connectives, quantifiers, and grouping symbols.

We use infix notation for the fixed predicate symbols $\triangleleft$, $\triangleleft^*$, $\triangleleft^+$, $\prec$, and $\approx$.

The rank of the variables determines the arity of the relations they range over. Variables of rank 0 range over individuals, those of rank 1 range over sets of individuals. The set $\boldsymbol{X}^i$ is the set of variables in $\boldsymbol{X}$ with rank i. We use lower-case for individual variables and constants and upper-case for set variables and predicate symbols. Further, we will say $X(x)$ to assert that the individual assigned to the variable x is included in the set assigned to the variable X. So, for instance,

$$(\forall y)[x \triangleleft^* y \rightarrow X(y)]$$

asserts that the set assigned to X includes every node dominated by the node assigned to x.

$L^2_{K,P}$ is a relational language, its terms are just the members of $\boldsymbol{K} \cup \boldsymbol{X}^0$. **Atomic formulae**, **literals** and **well-formed-formulae** are generated in the usual fashion. We use t, u, v, etc. to denote terms and ϕ, ψ, etc. to denote wffs. R denotes any of the predicates.

3.2 Tree Axioms

Let $\mathcal{A}^1_T$ denote the following set of axioms:

A1 $(\exists x)(\forall y)[x \triangleleft^* y]$,

A2 $(\forall x, y)[(x \triangleleft^* y \wedge y \triangleleft^* x) \rightarrow x \approx y]$,

A3 $(\forall x, y, z)[(x \triangleleft^* y \wedge y \triangleleft^* z) \rightarrow x \triangleleft^* z]$,

A4 $(\forall x, y)[x \triangleleft y \rightarrow$
$\qquad (x \triangleleft^+ y \wedge (\forall z)[(x \triangleleft^* z \wedge z \triangleleft^* y) \rightarrow (z \triangleleft^* x \vee y \triangleleft^* z)])]$,

A5 $(\forall x, z)[z \triangleleft^+ x \rightarrow (\exists y)[y \triangleleft x]]$,

A6 $(\forall x, z)[x \triangleleft^+ z \rightarrow (\exists y)[x \triangleleft y \wedge y \triangleleft^* z]]$,

A7 $(\forall x, y)[x \prec y \leftrightarrow (\neg x \triangleleft^* y \wedge \neg y \triangleleft^* x \wedge y \not\prec x)]$,

A8 $(\forall w, x, y, z)[(x \prec y \wedge x \triangleleft^* w \wedge y \triangleleft^* z) \rightarrow w \prec z]$,

A9 $(\forall x, y, z)[(x \prec y \wedge y \prec z) \rightarrow x \prec z]$,

A10 $(\forall x)[(\exists y)[x \triangleleft y] \rightarrow (\exists y)[x \triangleleft y \wedge (\forall z)[x \triangleleft z \rightarrow z \not\prec y]]]$,

A11 $(\forall x)[(\exists y)[x \prec y] \rightarrow (\exists y)[x \prec y \wedge (\forall z)[x \prec z \rightarrow z \not\prec y]]]$,

A12 $(\forall x)[(\exists y)[x \prec y] \rightarrow (\exists y)[x \prec y \wedge (\forall z)[x \prec z \rightarrow y \not\prec z]]]$.

The intuitive meaning of ***A1*** is that every tree includes a root which dominates every node in the tree. We will assume normal models (in which the interpretation of $\approx$ is fixed as equality in the domain of the model). ***A2*** then requires domination to be anti-symmetric. The anti-symmetry of domination implies that the root is unique. ***A3*** requires domination to be transitive. Reflexivity of domination follows from ***A7***. ***A4*** implies that there is no node that falls properly between, wrt domination, any node and its parent.

The axiom ***A5*** requires that every node other than the root has a parent. A **path** from x to y is the set of nodes that dominate y and are dominated by x. By ***A6***, every path from x that is non-trivial (includes some node other than x) includes a child of x. By ***A4*** this is minimal wrt domination among the nodes in the path properly dominated by x. **Linear branching** is a property of trees that requires every path to be linearly ordered by proper domination. (Here it is a consequence of ***A7*** and ***A8***, as we will show below.) Along with ***A5***, this implies that every non-trivial path ending at y includes the parent of y, which must be maximal wrt domination among the nodes in the path properly dominating y. If we restrict the class of models to those in which all paths are finite, then these axioms, with ***A3*** and ***A4***, fix proper domination as the transitive closure of parent.[2]

The forward ($\rightarrow$) direction of ***A7*** requires every pair of nodes to be related by either domination or left-of. This is sometimes referred to as the **exhaustiveness** property. The converse direction (**exclusiveness**) asserts that these relations are mutually exclusive (and that left-of is asymmetric). Together with ***A2*** these establish the fact that every tree is totally ordered by the relation $x \trianglelefteq y \overset{\text{def}}{\iff} x \triangleleft^* y \vee x \prec y$. (This gives the *depth-first* ordering of the tree.) ***A8*** is sometimes referred to as the **inheritance** (or **non-tangling**) property. It requires that the entire subtrees rooted at nodes related by left-of are also related by left-of. ***A9*** establishes transitivity of left-of. With ***A7*** this establishes that left-of linearly orders each set of siblings. ***A10***, then, requires that linear order to have a minimum and ***A11*** and ***A12*** require it to be discrete.

[2]Axioms $A5$ and $A6$ are only independent if we allow models with infinite paths.

As we shall see, these axioms by themselves do not suffice to define the class of variably branching trees. In fact a simple compactness argument shows that this class is not first-order axiomatizable (see Backofen et al. 1995). They do imply, though, most of the properties of trees usually encountered in the literature (for instance, McCawley 1968, Siefkes 1978, Radford 1988, Partee et al. 1990, Blackburn et al. 1993, Kayne 1994). Shortly, we will introduce two second-order induction axioms which we will show extend them sufficiently to capture exactly the structures we are interested in.[3] Thus $\mathcal{A}^1_T$ provides a sound and complete basis for reasoning about trees if one admits induction as a rule of inference.

One example of a common property of trees that we capture as a consequence rather than explicitly is linear branching. To see this, suppose that $x \triangleleft^* z \wedge y \triangleleft^* z$. By $\boldsymbol{A7}$ we have $x \triangleleft^* y \vee y \triangleleft^* x \vee x \prec y \vee y \prec x$. But, by $\boldsymbol{A8}$ and $\boldsymbol{A7}$, $x \prec y$ implies $z \prec y$ which, in turn, implies $\neg y \triangleleft^* z$, contradicting our hypothesis. A similar line of reasoning rules out $y \prec x$. Consequently, $x \triangleleft^* y \vee y \triangleleft^* x$.

3.3 Models and Satisfaction

Models are ordinary structures interpreting the individual constants and predicate symbols.

Definition 2 A **model** for the language $L^2_{K,P}$ is a tuple:

$$\langle \mathcal{U}, \mathcal{I}, \mathcal{P}, \mathcal{D}, \mathcal{L}, \mathcal{R}_p \rangle_{p \in P},$$

where:
$\mathcal{U}$ is a non-empty domain,
$\mathcal{I}$ is a function from $\boldsymbol{K}$ to $\mathcal{U}$,
$\mathcal{P}$, $\mathcal{D}$, and $\mathcal{L}$ are binary relations over $\mathcal{U}$
(interpreting $\triangleleft$, $\triangleleft^*$, and $\prec$ respectively),
$\mathcal{R}_p$ is a subset of $\mathcal{U}$ interpreting p, for each $p \in \boldsymbol{P}$.

If the domain of $\mathcal{I}$ is empty (i.e., the model is for a language $L_{\emptyset,P}$) we will generally omit it. Models for $L_{\emptyset,\emptyset}$, then, are tuples $\langle \mathcal{U}, \mathcal{P}, \mathcal{D}, \mathcal{L} \rangle$.

In general, satisfaction is relative to an assignment mapping each individual variable into a member of $\mathcal{U}$ and each predicate variable into a subset of $\mathcal{U}$. We use

$$M \models \phi\,[s]$$

to denote that a model M satisfies a formula ϕ with an assignment s. The notation

$$M \models \phi$$

[3] In Backofen et al. 1995 we show that the corresponding first-order schemas suffice to define the *theory* of these trees although not the class of structures.

asserts that M models ϕ with any assignment. When ϕ is a sentence we will usually use this form.

Proper domination is a defined predicate:

$$t \triangleleft^+ u \equiv t \triangleleft^* u \wedge \neg u \triangleleft^* t.^4$$

That the axioms $\mathcal{A}_T^1$ are consistent follows from the fact that they are satisfiable, at least by the structure consisting of a single node. It is easy to exhibit structures for each of the axioms which fail to satisfy that axiom while satisfying all of the others. Thus, they are independent as well.

If M is a model for a language $L^2_{K,P}$, then $\mathbf{Th}(M)$ is the set of sentences satisfied by M. If $\mathcal{M}$ is a set of models, then $\mathbf{Th}(\mathcal{M})$ is the set of sentences satisfied by all $M \in \mathcal{M}$. If Φ is a set of sentences, $\mathbf{Mod}(\Phi)$ is the set of models that satisfy each of the sentences in Φ and $\mathbf{Cn}(\Phi)$ is $\mathbf{Th}(\mathbf{Mod}(\Phi))$, the consequences of Φ.

3.3.1 Intended Models

For our standard definition of trees we adopt **tree domains** (Gorn 1967). A tree domain is, in essence, the set of node addresses in a tree in which the root has been given address ϵ and the children of the node at address w are given addresses (in order, left to right) $w \cdot 0, w \cdot 1, \ldots$, where $\cdot$ denotes concatenation.[5] Tree domains, then, are particular subsets of $\mathbb{N}^*$. ($\mathbb{N}$ is the set of natural numbers.)

Definition 3 A **tree domain** is a non-empty set $T \subseteq \mathbb{N}^*$, satisfying, for all $u, v \in \mathbb{N}^*$ and $i, j \in \mathbb{N}$, the conditions:

TD1 $\quad uv \in T \Rightarrow u \in T,$ $\qquad$ ***TD2*** $\quad ui \in T, j < i \Rightarrow uj \in T.$

Our intended models are those structures that, when restricted to $L_{\emptyset,\emptyset}$, are isomorphic to a tree domain under its natural interpretation.

Definition 4 The **natural interpretation** of a tree domain T is a model $T^\natural = \langle T, \mathcal{P}_T, \mathcal{D}_T, \mathcal{L}_T \rangle$, where:

$$\begin{aligned}
\mathcal{P}_T &= \{\langle u, ui\rangle \in T \times T \mid u \in \mathbb{N}^*, i \in \mathbb{N}\}, \\
\mathcal{D}_T &= \{\langle u, uv\rangle \in T \times T \mid u, v \in \mathbb{N}^*\}, \\
\mathcal{L}_T &= \{\langle uiv, ujw\rangle \in T \times T \mid u, v, w \in \mathbb{N}^*, i < j \in \mathbb{N}\}.
\end{aligned}$$

These are just those models of the axioms $\mathcal{A}_T^1$ for which the sets

$$\mathbf{B}_x = \{y \mid \langle y, x\rangle \in \mathcal{D}\}$$

and

$$\mathbf{L}_x = \{y \mid (\exists z)[\langle z, y\rangle, \langle z, x\rangle \in \mathcal{P} \text{ and } \langle y, x\rangle \in \mathcal{L}]\}$$

[4] We reserve the symbol $\equiv$ for *explicit* definitions. The left-hand side can be taken to be the syntactic equivalent of the right-hand side.

[5] We will usually dispense with the dot and denote concatenation by juxtaposition.

are finite, that is, for which the length of the path from the root to any node and the number of left siblings of any node are finite.

In these structures every **branch**—every maximal set of nodes that is linearly ordered by domination—is isomorphic to an initial subset of the natural numbers ordered by less-than-or-equals. Similarly, the set of children of any node is also isomorphic to such an initial segment of the natural numbers. Consequently, these models admit proofs by induction on the depth of a node in the tree and on the number of its left siblings. Note that, since every node has countably many children, there are countably many nodes at any given depth in the tree. Since the trees are countably deep, the domain of these structures is countable.

Lemma 1 *If $T \subseteq \mathbb{N}^*$ is a tree domain, then $T^\natural \models \mathcal{A}_T^1$ and for all $x \in T$, $\mathbf{B}_x$ and $\mathbf{L}_x$ are finite.*

The finiteness of $\mathbf{B}_x$ and $\mathbf{L}_x$ is clear. The proof, then, consists of verifying that $T^\natural$ satisfies each of the axioms. This is straightforward but tedious.

Lemma 2 *Suppose $M = \langle \mathcal{U}, \mathcal{P}, \mathcal{D}, \mathcal{L} \rangle$ is a model of $\mathcal{A}_T^1$ in which $\mathbf{B}_x$ and $\mathbf{L}_x$ are finite for all $x \in \mathcal{U}$. Then there is some tree domain $T \subseteq \mathbb{N}^*$ for which $T^\natural \cong M$ ($T^\natural$ is isomorphic to M).*

Proof. Let $l_M : \mathcal{U} \to \mathbb{N}^*$ be defined:

$$l_M(x) = \begin{cases} \epsilon & \text{if } (\forall y)[\langle y, x \rangle \notin \mathcal{P}], \\ l_M(y) \cdot i & \text{if } \langle y, x \rangle \in \mathcal{P} \text{ and } \mathbf{card}(\mathbf{L}_x) = i. \end{cases}$$

Let $l(M) = \boldsymbol{\rho}\, l_M$ (the range of l_M).

We claim that l_M is total, well-defined, and that $l(M)$ is a tree domain, i.e., a non-empty subset of $\mathbb{N}^*$ that satisfies conditions $\boldsymbol{TD1}$ and $\boldsymbol{TD2}$. It follows then, from the definitions of l_M and $l(M)^\natural$, that $l(M)^\natural \cong M$.

To establish the claim:

Since M is a model of $\mathcal{A}_T^1$, by $\boldsymbol{A1}$ there is some node in $\mathcal{U}$ that dominates every member of $\mathcal{U}$. By $\boldsymbol{A2}$ it is unique. Let $\mathcal{R}$ denote this node—the root of M. First, we show for every individual x in $\mathcal{U}$ except the root that there is a unique y such that $\langle x, y \rangle \in \mathcal{P}$, and that there is no $y \in \mathcal{U}$ such that $\langle y, \mathcal{R} \rangle \in \mathcal{P}$. (It follows then, that $l_T(\mathcal{R}) = \epsilon$, and thus, $\boldsymbol{\rho}\, l_T \neq \emptyset$.)

$\langle \mathcal{R}, x \rangle \in \mathcal{D}$ for all $x \in \mathcal{U}$ by $\boldsymbol{A1}$. Thus, by $\boldsymbol{A4}$ and the definition of $\lhd^+$, there is no $y \in \mathcal{U}$ such that $\langle y, \mathcal{R} \rangle \in \mathcal{P}$. Further, by $\boldsymbol{A1}$ and $\boldsymbol{A5}$ and linear branching, for all $x \in \mathcal{U}$ either $x = \mathcal{R}$ or there exists y such that $\langle \mathcal{R}, y \rangle \in \mathcal{D}$ and $\langle y, x \rangle \in \mathcal{P}$.

Uniqueness of the parent of any x follows from linear branching, $\boldsymbol{A4}$, and $\boldsymbol{A2}$ as follows: Suppose both y and x are parents of z. By $\boldsymbol{A4}$

they both properly dominate z. Then, by linear branching, one must dominate the other. But, by ***A4*** again, this implies they each dominate the other, and, by ***A2*** they must be equal.

With this we can now establish, by an induction on the depth of the node, that $l_M(x)$ has a unique value for every node $x \in \mathcal{U}$ and that $\boldsymbol{\rho} l_M$ satisfies ***TD1*** and ***TD2***. Note that $\mathbf{B}_x$ and $\mathbf{L}_x$ are defined for all $x \in \mathcal{U}$, and that $\mathbf{card}(\mathbf{B}_x) \geq 1$ for all such x. Suppose $\mathbf{card}(\mathbf{B}_x) = 1$. Then $x = \mathcal{R}$ and for all $y \in \mathcal{U}$, $\langle y, x\rangle \notin \mathcal{P}$. Thus, $l_M(x) = \epsilon$ and $\mathbf{L}_x = \emptyset$. Suppose $\mathbf{card}(\mathbf{B}_x) = n > 1$ and for all y if $\mathbf{card}(\mathbf{B}_y) < n$, then $l_M(y)\!\downarrow$. Since $\mathbf{card}(\mathbf{B}_x) \neq 1$, $x \neq \mathcal{R}$ and there is some unique y such that $\langle y, x\rangle \in \mathcal{P}$, and for that y, $\langle y, x\rangle \in \mathcal{D}$, and $\langle x, y\rangle \notin \mathcal{D}$.

For all $z \in \mathbf{B}_x$, $\langle z, x\rangle \in \mathcal{D}$. Consequently, by linear branching, either $\langle z, y\rangle \in \mathcal{D}$ or $\langle y, z\rangle \in \mathcal{D}$. By ***A4*** and ***A2***, then, either $\langle z, y\rangle \in \mathcal{D}$ or $z = x$. Therefore, $\mathbf{B}_y = \mathbf{B}_x \setminus \{x\}$, and $\mathbf{card}(\mathbf{B}_y) = n - 1$.

Thus, $l_M(y)\!\downarrow$ and $l_M(x) = l_M(y) \cdot \mathbf{card}(\mathbf{L}_x)$. Further, by definition of $\mathbf{L}_x$, for all $z \in \mathbf{L}_x$, $\langle y, z\rangle \in \mathcal{P}$ and $l_M(z) = l_M(y) \cdot \mathbf{card}(\mathbf{L}_z)$.

That $\boldsymbol{\rho} l_M$ satisfies ***TD1*** and ***TD2*** then follows immediately from the definition of l_M and the fact that it is total on $\mathcal{U}$. □

3.3.2 Induction Axioms

While each of our intended models satisfy $\mathcal{A}^1_T$, the converse is not true—there are structures that satisfy $\mathcal{A}^1_T$ but are not structures of our intended sort. As a result, the consequences of the axioms are a proper subset of the theory of trees. As far as that theory is concerned, the key distinction between these nonstandard models and the intended models is that, in the nonstandard models, induction on the depth of nodes (or on the number of left-siblings) is not valid. Consider the model M_1 of Figure 1. This consists of two components: an infinite sequence of nodes, each with a single child, extending up from the root; and, infinitely far out, a second component in which every node has exactly two children, every node has a parent in that component, and every node is dominated by every node in the first component.[6] It is easy to verify that this is a model of $\mathcal{A}^1_T$, although clearly $\mathbf{B}_x$ is infinite for every x in the second component. This model satisfies the sentence

$$(\forall x, y)[x \triangleleft y \wedge (\exists! z)[x \triangleleft z] \rightarrow (\exists! z)[y \triangleleft z]],$$

which says that every node whose parent has exactly one child also has exactly one child. Along with the fact that the root has exactly one child, this is sufficient to establish by induction, in standard models,

[6] This example is due to Backofen and Vijay-Shanker.

M_1

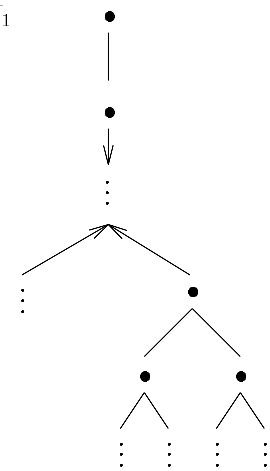

FIGURE 1 A nonstandard model of $\mathcal{A}_T^1$.

that every node has exactly one child. That is,

$$\begin{array}{l}((\exists x)[(\forall y)[x \triangleleft^* y] \wedge (\exists! y)[x \triangleleft y]] \wedge \\ \quad (\forall x, y)[(x \triangleleft y \wedge (\exists! z)[x \triangleleft z]) \rightarrow (\exists! z)[y \triangleleft z]]) \quad \rightarrow \quad (\forall x)(\exists! y)[x \triangleleft y]\end{array}$$

is valid in the set of standard models. On the other hand, it is clearly not satisfied in M_1.

To rule out such models we add two monadic second-order axioms. Let $\mathcal{A}_T$ denote $\mathcal{A}_T^1$ augmented with:

$$\boldsymbol{A}_{\mathrm{WF-D}} \qquad (\forall X)[(\exists x)[X(x)] \rightarrow (\exists x)[X(x) \wedge (\forall y)[y \triangleleft^+ x \rightarrow \neg X(y)]]$$

$$\boldsymbol{A}_{\mathrm{WF-L}} \qquad (\forall X)[(\exists x)[X(x)] \rightarrow (\exists x)[X(x) \wedge (\forall y)[y \prec x \rightarrow \neg X(y)]]$$

These axioms simply require proper-domination and left-of each to be well-founded, that is, there are no infinite sequences of nodes each of which properly dominates (respectively, is left-of) its predecessor. It is well known that well-foundedness of proper domination is equivalent to validity of induction on parent and similarly for left-of.[7] These axioms, then, restrict us to structures in which induction is valid—to our intended models.

[7] To see that well-foundedness implies induction, assume (for contradiction) that the root is in a set S and that S is closed under parent, but that $\bar{S}$ (the complement of S) is not empty. Consider, then, any node in $\bar{S}$ that is minimal wrt domination. For the other direction, assume that S is an infinite descending sequence of nodes but that induction on parent is valid. Consider the set of all nodes that are not dominated by any node in S. Clearly, the root is in this set and it is closed under parent. But just as clearly it contains no node in S.

Lemma 3 *In every model of $\mathcal{A}_T$ both $\mathbf{B}_x$ and $\mathbf{L}_x$ are finite for every x in the domain of the model.*

Proof. The proof is nearly immediate. Consider the set of all nodes for which $\mathbf{B}_x$ is infinite. There can be no node in this set that is minimal wrt domination, since the parent of any such minimal node would be dominated by only finitely many nodes. This set must, consequently, be empty. Similarly for $\mathbf{L}_x$. $\square$

Lemma 4 *Suppose M is a model of $\mathcal{A}_T^1$ and both $\mathbf{B}_x$ and $\mathbf{L}_x$ are finite for every x in the domain of M. Then M satisfies $\mathcal{A}_T$.*

Proof. Again the proof is almost immediate. Assume, for instance, that $\mathbf{B}_x$ is finite for all x and that S is non-empty. Choose $a \in S$. Then $\mathbf{B}_a \cap S$ is finite and non-empty and therefore contains an element that is minimal wrt domination. Such a point is, clearly, minimal in S as well. $\square$

Together with Lemmas 1 and 2 these prove the correctness of the axiomatization.

Theorem 5 *$M \models \mathcal{A}_T$ iff there is some tree domain $T \subseteq \mathbb{N}^*$ for which $T^\natural \cong M|_{L_{\emptyset,\emptyset}}$. (Where $M|_{L_{\emptyset,\emptyset}}$ is M restricted to the signature $L_{\emptyset,\emptyset}$.)*

4

$L^2_{K,P}$ and SnS

The signature of $L^2_{K,P}$ was chosen to directly express those relations within trees that typically occur in linguistic theories. We choose to work with the monadic second-order language over that signature because, while this is a relatively expressive language, it is still *solvable*—it is decidable whether a given formula in the language is satisfiable.[1] We establish this, in this chapter, by reducing satisfiability in $L^2_{K,P}$ to membership in SnS—the monadic second-order theory of multiple successor functions. This is the theory of $\mathcal{N}_n$, the complete n-branching tree, and the structures we are interested are a definable class of subsets of $\mathcal{N}_\omega$. Thus, the question of whether any tree satisfies a given formula ϕ: "$\mathbf{Mod}(\phi) \neq \emptyset$?", becomes the question of whether there is a $\phi'(X)$, a suitable relativization of ϕ to X, for which "$(\exists X)[\phi'(X)] \in$ SωS $(= \mathbf{Th}(\mathcal{N}_\omega))$?". We actually get a stronger result—not only can theories expressed in $L^2_{K,P}$ be reduced to fragments of SnS, but the converse holds as well, SnS can be interpreted in $L^2_{K,P}$. Thus, in a strong sense, $L^2_{K,P}$ and SnS are equivalent.

As it turns out, many of the properties and relationships over individuals and over trees that we are interested in can be expressed in $L^2_{K,P}$, equivalently in SnS, but not all. The issue of what can and can not be expressed is a fruitful one, and it is this question that leads us, ultimately, to the primary result of this part—the characterization of the class of strongly context-free languages in terms of definability in $L^2_{K,P}$.

[1] In fact, it seems likely that $L^2_{K,P}$ is very nearly maximally expressive among solvable languages over this signature.

4.1 SnS

For $n \in I\!N \cup \{\omega\}$, let $T_n = \boldsymbol{n}^*$, where $\boldsymbol{n} = \{i \mid i < n\}$. T_n is the n-branching tree-domain, i.e., the tree-domain in which every address has n successors. In keeping with our interpretation of tree-domains, we have, for all $x, y \in T_n$, that $x \triangleleft^* y \Leftrightarrow y = xz$ for some $z \in \boldsymbol{n}^*$. Let $\trianglelefteq$ denote the lexicographic order on T_n:

$$x \trianglelefteq y \overset{\text{def}}{\Longleftrightarrow} x \triangleleft^* y \text{ or } x = zau,\ y = zbv,\ \text{ and } a < b$$

for $x, y, z, u, v \in \boldsymbol{n}^*$ and $a, b \in \boldsymbol{n}$. Let r_i denote the i^{th} successor function (which we will generally use in post-fix position): $xr_i = xi$, for $x \in T_n$. Then, again for $n \in I\!N \cup \{\omega\}$,

$$\mathcal{N}_n \overset{\text{def}}{=} \langle T_n, \triangleleft^*, \trianglelefteq, r_i \rangle_{i<n}$$

is the **structure of n successor functions**. (We conflate $\triangleleft^*$, $\trianglelefteq$, and r_i with their interpretation in the structure.) The **monadic second-order theory of $\mathcal{N}_n$** is

$$\mathrm{S}n\mathrm{S} \overset{\text{def}}{=} \mathbf{Th}_2(\mathcal{N}_n).$$

Note that the language of SnS contains unary functions but no constants.

Rabin's fundamental result is that S2S is decidable. (It follows, by a reasonably easy interpretation, that SnS for every $n \leq \omega$ is decidable as well.) The proof involves a reduction of the problem to the emptiness problem for a class of automata on infinite trees. We will not discuss it in detail here, but as we will need to appeal to the automata later, we will describe them.

4.2 Automata on Infinite Trees

Definition 1 If Σ is any finite alphabet, an **n-ary Σ-valued tree** is a map $T^n_\Sigma : T_n \to \Sigma$.

Thus, an n-ary Σ-valued tree is just the tree T_n with the nodes labeled with elements of Σ. In sequel, unless stated otherwise, we will assume binary Σ-valued trees which we will denote T_Σ.

Definition 2 A **Rabin Tree Automaton** over binary Σ-valued trees is a tuple $\mathcal{A} = \langle Q, q_0, \Delta, F \rangle$ where:

- Q is a finite set of **states**,
- $q_0 \in Q$ is the **start state**,
- $\Delta : Q \times \Sigma \to \mathcal{P}(Q \times Q)$ is a (non-deterministic) **transition function**,
- and $F \subseteq \mathcal{P}(Q)$ is the set of **accepting subsets**.[2]

[2]There are a number of variations on the definition of acceptance. In **Büchi au-**

Definition 3 A **branch**[3] π in a tree T_n is a maximal subset of T_n that is linearly ordered by $\triangleleft^*$.

Definition 4 A **run** of an automata $\mathcal{A}$ over Σ on a Σ-valued tree T_Σ is a map $r : T_2 \to Q$ in which $r(\varepsilon) = q_0$ and $\langle r(x0), r(x1)\rangle \in \Delta(r(x), T_\Sigma(x))$ for all $x \in T_n$.

Definition 5 A run r of $\mathcal{A}$ on T_Σ is **accepting** iff

$$\mathrm{In}(r|_\pi) \in F, \text{for all branches } \pi.$$

Where $\mathrm{In}(f) \stackrel{\text{def}}{=} \{b \in B \mid (\exists^\infty a \in A)[f(a) = b]\}$ for any map $f : A \to B$, and $r|_\pi$ is r restricted to π.

A Σ-valued tree T_Σ is accepted by an automaton $\mathcal{A}$ iff there is an accepting run of $\mathcal{A}$ on T_Σ. A set of trees is **Rabin recognizable** iff it is the set of trees accepted by some Rabin tree automaton.

The operation of an automaton, as with (top-down) tree automata over finite trees, can be thought of as starting with a single automaton in state q_0 at the root and proceeding by sending automata in states q' and q'' to the 0 and 1 successor of node w if w is labeled a, the automaton at w is in state q and $\langle q', q''\rangle \in \Delta(q, a)$.

Rabin's theorem follows from the fact that there is an effective procedure for constructing, from any formula $\phi(X_1, \ldots, X_n)$ in the language of S2S in which the free variables are among the X_i, an automaton $\mathcal{A}_\phi$ over $\{0,1\}^n$-valued trees such that a tree T is accepted by $\mathcal{A}_\phi$ iff the assignment

$$s_T \stackrel{\text{def}}{=} \{X_i \mapsto \{w \mid \boldsymbol{\pi}_i(T(w)) = 1\} \mid i \leq n\}$$

(where $\boldsymbol{\pi}_i$ is the i^{th} projection) is a satisfying assignment for ϕ, i.e.,

$$\mathcal{N}_2 \models \phi(X_1, \ldots, X_n)\,[s_T].$$

Thus, the set of trees accepted by the automaton is non-empty iff the formula is satisfiable. Rabin shows that emptiness of the language accepted by these automata is decidable, and the decidability of S2S follows.

tomata, which Rabin calls **weak automata**, F is just a subset of Q. Thomas (1990), refers to the automata we define here as **Muller automata**. He defines Rabin automata as an equivalent variation in which acceptance is defined by a sequence of pairs of finite subsets of Q, $(\langle L_i, U_i\rangle)_{i<n}$, in which the members of L_i may not occur infinitely often along any path while some member of U_i must occur infinitely often along every path. Rabin, on the other hand, uses the definition we give in Rabin 1969 and refers to the variation as Muller automata in Rabin 1972.

[3] In the SnS literature this is referred to as a *path*, but we prefer to use path for a sequence of nodes falling between two nodes wrt domination. Such a sequence, in turn, is usually referred to as a *chain* but this conflicts with use of the term in GB.

4.3 Interpreting SnS in $L^2_{\emptyset,\emptyset}$

Recall that $L^2_{\emptyset,\emptyset}$ denotes the monadic second-order language with no individual constants or predicate symbols other than the fixed predicate symbols $\triangleleft$, $\triangleleft^*$, $\triangleleft^+$, $\prec$, and $\approx$. The purpose of this section is to show that, roughly speaking, anything that can be said in the language of SnS can be said in $L^2_{\emptyset,\emptyset}$. That is to say, there is a translation from formulae in the language of SnS into $L^2_{\emptyset,\emptyset}$ such that a formula is in SnS iff its translation is in $\mathbf{Th}_2(T_n^\natural)$ where $T_n^\natural$ is the natural interpretation of T_n from Definition 4 of Section 3.3.1. Such a translation is referred to as a **faithful interpretation** of SnS in $\mathbf{Th}_2(T_n^\natural)$ (see Enderton 1972, §2.7). Note that we are dealing with the monadic second-order theory here, so assignments map individual variables to elements of T_n and predicate variables to subsets of T_n. Since $\mathcal{N}_n$ and $T_n^\natural$ share the same universe, assignments for the former serve for the latter as well. We give our interpretation with a sequence of assertions that determine a syntactic translation of formulae in the language of SnS into $L^2_{\emptyset,\emptyset}$. The assertions form the core of a proof that the translation does in fact give a faithful interpretation. This and the proofs of the assertions are reasonably self-evident, and we don't give them here.

We begin by eliminating function symbols. Let $\phi(tr_i)$ denote any formula in which the term tr_i, for some term t, occurs and let $\phi(x)$, in the same context, denote $\phi(tr_i)$ with x replacing every occurrence of tr_i.

$$\begin{aligned}
\mathcal{N}_n \models \phi(tr_i)\,[s] \quad &\Leftrightarrow \quad \mathcal{N}_n \models (\exists x)[x \approx tr_i \wedge \phi(x)]\,[s] \\
\mathcal{N}_n \models x \approx tr_0\,[s] \quad &\Leftrightarrow \quad \mathcal{N}_n \models (t \triangleleft^* x \wedge (\forall y)[(t \triangleleft^* y \wedge t \not\approx y) \rightarrow x \preceq y])\,[s] \\
\mathcal{N}_n \models x \approx tr_i\,[s] \quad &\Leftrightarrow \quad \mathcal{N}_n \models (t \triangleleft^* x \wedge (\forall y)[tr_{i-1} \prec y \rightarrow x \preceq y])\,[s],\ i > 0.
\end{aligned}$$

As an example

$$\begin{aligned}
&\mathcal{N}_n \models x \approx t2\,[s] \Leftrightarrow \\
&\quad \mathcal{N}_n \models (\exists x_0)[t \triangleleft^* x_0 \wedge (\forall y)[(t \triangleleft^* y \wedge t \not\approx y) \rightarrow x_0 \preceq y] \wedge \\
&\qquad\qquad (\exists x_1)[t \triangleleft^* x_1 \wedge (\forall y)[x_0 \prec y \rightarrow x_1 \preceq y] \wedge \\
&\qquad\qquad\qquad t \triangleleft^* x \wedge (\forall y)[x_1 \prec y \rightarrow x \preceq y]]]\,[s].
\end{aligned}$$

The translation introduces i new individual variables for each tr_i, but, while r_i is unbounded in general, the maximum r_i occurring in any finite ϕ is finite. Thus, the translation of ϕ is finite as well.

The actual translation into $L^2_{\emptyset,\emptyset}$ is induced by the translation of first-order atomic formulae. The second-order atomic formulae need no translation.

$$\begin{aligned}
\mathcal{N}_n \models x \triangleleft^* y\,[s] \quad &\Leftrightarrow \quad T_n^\natural \models x \triangleleft^* y\,[s] \\
\mathcal{N}_n \models x \preceq y\,[s] \quad &\Leftrightarrow \quad T_n^\natural \models x \prec y \vee x \triangleleft^* y\,[s].
\end{aligned}$$

In our example

$$
\begin{aligned}
\mathcal{N}_n \models x \approx t2\,[s] \Leftrightarrow& \\
T_n^{\natural} \models (\exists x_0)[t \lhd^* x_0 \wedge (\forall y)[(t \lhd^* y \wedge t \not\approx y) &\rightarrow (x_0 \prec y \vee x_0 \lhd^* y)] \wedge \\
(\exists x_1)[t \lhd^* x_1 \wedge (\forall y)[x_0 \prec y &\rightarrow (x_1 \prec y \vee x_1 \lhd^* y)] \wedge \\
t \lhd^* x \wedge (\forall y)[x_1 \prec y &\rightarrow (x \prec y \vee x \lhd^* y)]]]\,[s].
\end{aligned}
$$

This translation, then, witnesses the theorem:

Theorem 1 *There is a syntactic translation $\phi \mapsto \phi'$ taking formulae in the language of* SnS *to formulae in $L^2_{\emptyset,\emptyset}$ such that, for all $n \leq \omega$, $\phi \in$* SnS *iff $\phi' \in \mathbf{Th}_2(T_n^{\natural})$.*

4.4 Defining Sets of Labeled Trees in SωS

The objects we are interested in describing are (generally finite) trees with variable branching in which individual nodes may be named by constants (in $\boldsymbol{K}$) and are labeled via (monadic) predicate symbols (in $\boldsymbol{P}$). The structure $\mathcal{N}_\omega$, in contrast, is infinite, has fixed branching, and interprets no parameters other than $\lhd^*$, $\underline{\prec\mkern-14mu\lhd}$, and the r_i. Our concern in this section is the embedding of our intended structures in $\mathcal{N}_\omega$. As suggested earlier, we can capture trees that are smaller than T_ω as suitably formed subsets. Thus, we are looking for a translation of a formula $\phi \in L^2_{K,P}$ into a formula $\phi'(X)$ in the language of SωS such that the trees in $\mathbf{Mod}(\phi)$ correspond to the satisfying assignments for X in $\phi'(X)$ in SωS. Within this framework, constants and predicate symbols in $\boldsymbol{K}$ and $\boldsymbol{P}$ can be interpreted as existentially quantified individual and set variables, respectively. As with the prior section, rather than lay out the translation explicitly, we make a sequence of assertions that both serves to define the translation and forms the basis of a proof of its correctness.

Theorem 2 *There is a translation $\phi \mapsto \phi'$ taking formulae in $L^2_{K,P}$ to formulae in the language of* SωS *such that ϕ is satisfiable over trees iff $\phi' \in$* SωS*.*

Suppose $\phi \in L^2_{K,P}$. We must show that there is a translation $\phi'(X)$ such that there is some tree M and assignment s that satisfy ϕ iff there is some s' for $\mathcal{N}_\omega$ that satisfies $\phi'(X)$.

Suppose $M = \langle \mathcal{U}, \mathcal{I}, \mathcal{P}, \mathcal{D}, \mathcal{L}, \mathcal{R}_P \rangle_{P \in \boldsymbol{P}}$ is a tree and $M \models \phi\,[s]$. Let $\boldsymbol{X}^0$ and $\boldsymbol{X}^1$ denote sets of individual and set variables, respectively, in $\boldsymbol{X}$, and suppose $\boldsymbol{X} = \boldsymbol{X}^0 \cup \boldsymbol{X}^1$. Then s maps $\boldsymbol{X}^0 \rightarrow \mathcal{U}$ and $\boldsymbol{X}^1 \rightarrow \mathcal{P}(\mathcal{U})$.

Let $\boldsymbol{P}_\phi$ and $\boldsymbol{K}_\phi$ denote the (finitely many) parameters that actually occur in ϕ.

By Lemma 2 of Section 3.3.1 there is a mapping $l : \mathcal{U} \rightarrow T_\omega$ such that $l(M)$ (the range of l) is a tree-domain.

For the forward direction, let

$$\begin{aligned} s' : \boldsymbol{X} \to T_\omega \quad = \quad & \{x \mapsto l(s(x)) \mid x \in \boldsymbol{X}^0\} \cup \\ & \{X \mapsto l(s(X)) \mid X \in \boldsymbol{X}^1\} \cup \\ & \{x_a \mapsto l(\mathcal{I}(a)) \mid a \in \boldsymbol{K}\} \cup \\ & \{X_P \mapsto l(\mathcal{R}_P) \mid P \in \boldsymbol{P}\} \cup \\ & \{X_{\mathcal{U}} \mapsto l(M)\} \end{aligned}$$

where the x_a, X_P, and $X_{\mathcal{U}}$ are new variables.

Then

$$\begin{aligned} M \models x \triangleleft^* y\,[s] \quad &\Leftrightarrow \quad \mathcal{N}_\omega \models x \triangleleft^* y\,[s'] \\ M \models x \prec y\,[s] \quad &\Leftrightarrow \quad \mathcal{N}_\omega \models x \underline{\triangleleft} y \wedge \neg x \triangleleft^* y\,[s'] \\ M \models x \triangleleft y\,[s] \quad &\Leftrightarrow \quad \mathcal{N}_\omega \models x \triangleleft^* y \wedge \\ & \qquad\qquad (\forall z)[(z \triangleleft^* y \wedge z \not\approx y) \to z \triangleleft^* x]\,[s'] \\ M \models P(x)\,[s] \quad &\Leftrightarrow \quad \mathcal{N}_\omega \models X_P(x)\,[s'] \\ M \models \phi(a)\,[s] \quad &\Leftrightarrow \quad \mathcal{N}_\omega \models \phi'(x_a)\,[s'] \\ M \models (\forall x)[\phi(x)]\,[s] \quad &\Leftrightarrow \quad \mathcal{N}_\omega \models (\forall x)[X_{\mathcal{U}}(x) \to \phi'(x)]\,[s'] \\ M \models (\forall X)[\phi(X)]\,[s] \quad &\Leftrightarrow \quad \mathcal{N}_\omega \models (\forall X)[(\forall x)[X(x) \to X_{\mathcal{U}}(x)] \to \\ & \qquad\qquad \phi'(X)]\,[s']. \end{aligned}$$

To get the other direction, i.e., that the existence of s' implies the existence of M and s, we must insure that the interpretation of $X_{\mathcal{U}}$ is a tree-domain, that the interpretation of each x_a is in the interpretation of $X_{\mathcal{U}}$ and that the interpretation of X_P is a subset of the interpretation of $X_{\mathcal{U}}$. Thus, we translate ϕ as $\phi'(X_{\mathcal{U}}, X_P, x_a)_{P \in P_\phi, a \in K_\phi}$, which is the conjunction of the translation of ϕ sketched above with:

$$\mathrm{Tree}(X_{\mathcal{U}}) \wedge \bigwedge_{a \in K_\phi} X_{\mathcal{U}}(x_a) \wedge \bigwedge_{P \in P_\phi} (\forall x)[X_P(x) \to X_{\mathcal{U}}(x)].$$

Where

$$\begin{aligned} &\mathrm{Tree}(X) \equiv \qquad\qquad (1) \\ &\quad (\exists x)(\forall y)[X(x) \wedge (X(y) \to x \triangleleft^* y)] \wedge \\ &\quad (\forall x, y, z)[(X(x) \wedge X(y) \wedge x \triangleleft^* z \wedge z \triangleleft^* y) \to X(z)] \wedge \\ &\quad (\forall w, x, y, z)[\\ &\qquad (X(x) \wedge X(y) \wedge w \triangleleft x \wedge w \triangleleft y \wedge w \triangleleft z \wedge x \prec z \wedge z \prec y) \to X(z)] \end{aligned}$$

(Recall that $\equiv$ denotes syntactic equivalence, that is $\mathrm{Tree}(X)$ is to be read as an abbreviation for the right hand side.)

First conjunct requires existence of a root, second requires X to be connected by parent, and the third requires the sets of siblings in X to be connected by immediate left-of.

This suffices to insure, for any s' such that

$$\mathcal{N}_\omega \models \phi'(X_\mathcal{U}, X_P, x_a)_{P \in P_\phi, a \in K_\phi}\,[s'],$$

that

$$\langle s'(X_\mathcal{U}), \mathcal{I}', \mathcal{D}', \mathcal{P}', \mathcal{L}', s'(X_P)\rangle_{P \in P_\phi} \models \phi\,[s]$$

where $\mathcal{I}' : a \mapsto s'(x_a)$, $\mathcal{D}'$, $\mathcal{P}'$, $\mathcal{L}'$ are interpreted as indicated above, and $s = s'|_X$.

If we work with SnS for some $n < \omega$ rather than SωS, then this translation serves to decide if ϕ is satisfiable over trees with branching less-than-or-equal to n. It is trivial, of course, to restrict this further to models with branching fixed at n. Then, taking $\boldsymbol{K} = \boldsymbol{P} = \emptyset$, Theorem 2 gives us the converse of Theorem 1.

Corollary 3 *There is a translation $\phi \mapsto \phi'$ taking formulae in $L^2_{\emptyset,\emptyset}$ to formulae in the language of* SnS *such that, for all $n \leq \omega$, $\phi \in \mathbf{Th}_2(T_n^\natural)$ iff $\phi' \in$* SnS.

4.4.1 Labeled Trees in $\mathcal{N}_\omega$

Labeled trees, then, in the context of SωS can be equated with the assignment for $X_\mathcal{U}$, X_P, and x_a.

Definition 6 A **labeled tree** in $\mathcal{N}_\omega$ (or just a *tree*) is an assignment:

$$M_1 = [X_\mathcal{U} \mapsto \mathcal{U}^1, X_i \mapsto X_i^1, x_j \mapsto x_j^1]_{i \leq n_1, j \leq m_1}$$

where

$$\begin{gathered} \mathcal{N}_\omega \models \mathrm{Tree}(X_\mathcal{U})[X_\mathcal{U} \mapsto \mathcal{U}^1], \\ X_i^1 \subseteq \mathcal{U}^1, i \leq n_1, \text{ and} \\ x_j^1 \in \mathcal{U}^1, j \leq m_1. \end{gathered}$$

Definition 7 A tree M_1 satisfies a formula ϕ iff it is a satisfying assignment for the variables in ϕ in $\mathcal{N}_\omega$.

Thus, it only satisfies formulae $\phi(X_\mathcal{U}, X_i, x_j)_{i \leq n_1, j \leq m_1}$ in which the free variables are among $\{X_\mathcal{U}, X_i, x_j \mid i \leq n_1, j \leq m_1\}$. Formally,

$$M_1 \models \phi(X_\mathcal{U}, X_i, x_j)_{i \leq n_1, j \leq m_1} \Leftrightarrow \mathcal{N}_\omega \models \phi(X_\mathcal{U}, X_i, x_j)_{i \leq n_1, j \leq m_1} M_1.$$

If $S \subseteq \mathcal{U}^1$ and $v \in \mathcal{U}^1$, then we will say

$$\begin{aligned} M_1 \models \phi(S, v) \quad &\text{iff} \quad M_1 \models \phi(X, x)[X \mapsto S, x \mapsto v] \\ &\text{iff} \quad \mathcal{N}_\omega \models \phi(X, x)\ M_1 \cup [X \mapsto S, x \mapsto v] \end{aligned}$$

where X and x are variables that are not in $\boldsymbol{\delta}\, M_1$ (the domain of M_1).

Definition 8 A formula $\phi(X)$ is **relativized** to X iff all quantification in $\phi(X)$ is relative to X, e.g.,

$$\begin{array}{l}(\exists x)[X(x) \wedge \cdots \\ (\forall x)[X(x) \rightarrow \cdots \\ (\exists Y)[\text{Subset}(Y, X) \wedge \cdots \\ (\forall Y)[\text{Subset}(Y, X) \rightarrow \cdots\end{array}$$

Let $\mathcal{A}_{\mathcal{U}}$ be $\mathcal{A}_T$, the axioms of Section 3.2, relativized to $X_{\mathcal{U}}$. Assignments into T_ω that satisfy $\mathcal{A}_{\mathcal{U}}$, even those in which the range of the assignment is the value of $X_{\mathcal{U}}$ under the assignment, do not quite correspond to trees in $\mathcal{N}_\omega$. This is because we have no "immediate left-of" predicate corresponding to the parent predicate. Thus, formulae that are relativized to some subset of T_ω are insensitive to children that the subset excludes. Thus there is no way for $\mathcal{A}_{\mathcal{U}}$ to enforce the third conjunct of Tree($X_{\mathcal{U}}$); subsets of T_ω in which sets of siblings are not necessarily connected wrt immediate left-of (in $\mathcal{N}_\omega$) are trees so far as $\mathcal{A}_T$ is concerned. Nonetheless, for convenience in working with trees in $\mathcal{N}_\omega$ directly in SωS (or rather, $L^2_{K,P}$) we would like the trees to exhibit this property.

Let $\boldsymbol{A}'(X_{\mathcal{U}})$, then, be the third conjunct of Tree($X_{\mathcal{U}}$) and let $\mathcal{A}^+_{\mathcal{U}}$ be $\mathcal{A}_{\mathcal{U}}$ extended with $\boldsymbol{A}'(X_{\mathcal{U}})$. Then, just as $\mathcal{A}_T$ characterizes the set of trees (among models with finite $\mathbf{B}_x$ and $\mathbf{L}_x$), $\mathcal{A}^+_{\mathcal{U}}$ characterizes the set of trees in $\mathcal{N}_\omega$.

Theorem 4 *If $M_1 = [X_{\mathcal{U}} \mapsto \mathcal{U}^1, X_i \mapsto X^1_i, x_j \mapsto x^1_j]_{i \leq n_1, j \leq m_1}$ is an assignment into T_ω in which $X^1_i \subseteq \mathcal{U}^1$, for $i \leq n_1$ and $x^1_j \in \mathcal{U}^1$, for $j \leq m_1$, then M_1 is a tree in $\mathcal{N}_\omega$ iff $M_1 \models \mathcal{A}^+_{\mathcal{U}}$.*

Proof. The proof is nearly immediate. Each of $\boldsymbol{A_2}(X_{\mathcal{U}})$ through $\boldsymbol{A_4}(X_{\mathcal{U}})$ and $\boldsymbol{A_7}(X_{\mathcal{U}})$ through $\boldsymbol{A_{12}}(X_{\mathcal{U}})$ are properties of every subset of T_ω. (This is easy to verify.) $\boldsymbol{A_1}(X_{\mathcal{U}})$ is just the first conjunct of Tree($X_{\mathcal{U}}$); and $\boldsymbol{A_5}(X_{\mathcal{U}})$ and $\boldsymbol{A6}(X_{\mathcal{U}})$ are (each) equivalent to the second conjunct of Tree($X_{\mathcal{U}}$). □

4.4.2 Isomorphisms between Trees in $\mathcal{N}_\omega$

Definition 9 If M_1 and M_2 are trees, then M_1 and M_2 are **isomorphic**, $M_1 \cong M_2$ iff there is a bijection $h : \mathcal{U}^1 \to \mathcal{U}^2$ such that:[4]

$$\begin{aligned} M_1 \models v \triangleleft^* w &\quad\Leftrightarrow\quad M_2 \models h(v) \triangleleft^* h(w) \\ M_1 \models v \prec w &\quad\Leftrightarrow\quad M_2 \models h(v) \prec h(w) \\ M_1 \models v \triangleleft w &\quad\Leftrightarrow\quad M_2 \models h(v) \triangleleft h(w) \end{aligned}$$
$$X_i^2 = \left\{ h(v) \mid v \in X_i^1 \right\}$$
$$x_j^2 = h(x_j^1).$$

Lemma 5 *If $M_1 \cong M_2$ and $\phi(X_{\mathcal{U}}, X_i, x_j)$ is relativized to $X_{\mathcal{U}}$, then*

$$M_1 \models \phi(X_{\mathcal{U}}, X_i, x_j) \Leftrightarrow M_2 \models \phi(X_{\mathcal{U}}, X_i, x_j).$$

For the first-order fragment, this is the homomorphism theorem. The second-order fragment is an easy extension.

[4] More properly, this would be

$$\mathcal{N}_\omega \models x \triangleleft^* y\ M_1 \cup [x \mapsto v, y \mapsto w] \Leftrightarrow \mathcal{N}_\omega \models x \triangleleft^* y\ M_2 \cup [x \mapsto h(v), y \mapsto h(w)]$$

and so on.

5

Definability and Non-Definability in $L^2_{K,P}$

5.1 Definability on a Class of Structures

We now turn to the issue of what constitutes a definition of a relation or a property with respect to a fixed class of structures (in our case the class of intended models (trees) for $L^2_{K,P}$ for some $\boldsymbol{K}$ and $\boldsymbol{P}$, or, in the degenerate case, $\mathcal{N}_\omega$). We are concerned here with both first- and **second-order relations**, by which we mean relations on predicates. We will confine our second-order relations, however, to relations on sets of individuals—those in which the arguments are monadic. This is sufficient for our purposes and simplifies the exposition greatly. We can admit hybrid relations in which some arguments range over individuals and others over sets of individuals by interpreting them as second-order relations in which some arguments are restricted to range over singletons. An **n-ary relation on a class of structures** $\mathcal{C}$ is an n-ary relation that is uniformly defined on the universes of the structures in $\mathcal{C}$. Following Gurevich (1988) we formalize this as a function $\boldsymbol{\mathcal{R}}$ taking each $M \in \mathcal{C}$ into, in the elementary case, a subset of $|M|^n$ (where $|M|$ is the universe of M). For a second-order relation $\boldsymbol{\mathcal{R}}(M)$ is a subset of $\mathcal{P}(|M|)^n$, the set of n-tuples of subsets of $|M|$.

In defining second-order relations we will have occasion to employ variables that range over relations between sets of individuals. We will be working, then, with three types of languages. For a given structure M, let L^1_M and L^2_M denote the first- and full (unrestricted arity) second-order languages for M. Let L^3_M denote the language L^2_M augmented with a set of third-order variables. We will use boldface type for the names of these variables and will continue to use lowercase for individual variables and uppercase for set variables. We will say $\boldsymbol{X}(X)$ to assert that the set of individuals assigned to X occurs in the set of sets assigned to

$\boldsymbol{X}$. In our applications third-order variables will never be bound by quantifiers and will always range over relations between sets. We can provide semantics for formulae in L^3_M, therefore, simply by extending the notion of an assignment appropriately.

For $\mathcal{C}$, a class of structures, let

$$L^i_{\mathcal{C}} = \bigcap_{M \in \mathcal{C}} [L^i_M],$$

for $1 \leq i \leq 3$.

We will use $\vec{x}_n \stackrel{\text{def}}{=} \langle x_1, \ldots, x_n \rangle$ to denote a sequence of distinct variables. For any assignment s, let $s(\vec{x}_n) = \langle s(x_1), \ldots, s(x_n) \rangle$. Similarly for $\vec{X}_n$, $s(\vec{X}_n)$, $\vec{\boldsymbol{X}}_n$ and $s(\vec{\boldsymbol{X}}_n)$.

5.1.1 Explicit Definitions

Definition 1 [Elementary definability] A relation $\boldsymbol{\mathcal{R}}(M) \subseteq |M|^n$ is **(explicitly) elementary definable** on a class of structures $\mathcal{C}$ iff there is a formula $\phi(\vec{x}_n)$ in the language $L^1_{\mathcal{C}}$ such that, for each $M \in \mathcal{C}$,

$$\mathcal{R}(M) = \{s(\vec{x}_n) \mid M \models \phi(\vec{x}_n)\,[s]\}\,.$$

Equivalently, if R is a predicate constant of arity n, not interpreted by M, and M' is any structure expanding M with an interpretation $\mathcal{R}'$ of R, then

$$M' \models (\forall \vec{x}_n)[R(\vec{x}_n) \leftrightarrow \phi(\vec{x}_n)] \text{ iff } \mathcal{R}' = \mathcal{R}.$$

The formula $R(\vec{x}_n) \leftrightarrow \phi(\vec{x}_n)$, then, is an (explicit) elementary definition of $\mathcal{R}$. Given such a definition, one can work in the language $L^1_{\mathcal{C}}$ expanded with the predicate R without actually leaving $L^1_{\mathcal{C}}$, since the definition can be regarded as a purely syntactic definition $R(\vec{x}_n) \equiv \phi(\vec{x}_n)$, that is, one can obtain an $L^1_{\mathcal{C}}$ formula by replacing every occurrence of $R(\vec{x}_n)$ in a formula with its definition $\phi(\vec{x}_n)$.

A relation on a class of structures is second-order definable iff it can be expressed in the second-order language over that class. The notion is meaningful for both elementary and second-order relations (as well as hybrid relations).

Definition 2 [Second-order definability] A relation $\boldsymbol{\mathcal{R}}(M) \subseteq |M|^n$ is **(explicitly) second-order definable** on a class of structures $\mathcal{C}$ iff there is a formula $\phi(\vec{x}_n)$ in the language $L^2_{\mathcal{C}}$ such that, for all $M \in \mathcal{C}$,

$$\boldsymbol{\mathcal{R}}(M) = \{s(\vec{x}_n) \mid M \models \phi(\vec{x}_n)\,[s]\}\,.$$

A second-order relation $\boldsymbol{\mathcal{R}}(M) \subseteq \mathcal{P}(|M|)^n$ is (explicitly) second-order definable on a class of structures $\mathcal{C}$ iff there is a formula $\phi(\vec{X}_n)$ in the language $L^2_{\mathcal{C}}$ such that, for all $M \in \mathcal{C}$,

$$\boldsymbol{\mathcal{R}}(M) = \left\{s(\vec{X}_n) \mid M \models \phi(\vec{X}_n)\,[s]\right\}.$$

Again, for our purposes, the value of these definitions is that we can employ second-order definable predicates freely without exceeding the expressive power of $L^2_{\mathcal{C}}$.

5.1.2 Implicit Definitions

Definition 3 [Implicit Elementary Definability] A relation $\boldsymbol{\mathcal{R}}(M) \subseteq |M|^n$ is **implicitly elementary definable** on a class of structures $\mathcal{C}$ iff there is a formula $\phi(X)$ in the language $L^2_{\mathcal{C}}$ in which X is the only second-order variable that occurs and only X occurs free, such that, for all $M \in \mathcal{C}$,

$$M \models \phi(X)[s] \Leftrightarrow \boldsymbol{\mathcal{R}}(M) = s(X).$$

It is more usual to take X to be an otherwise uninterpreted predicate symbol R. The definition ϕ is then a sentence and we require that, for all $M \in \mathcal{C}$ and all structures M' expanding M with an interpretation $\mathcal{R}'$ of R,

$$M' \models \phi \text{ iff } \mathcal{R}' = \mathcal{R}.$$

Definition 4 [Simultaneous Implicit Elementary Definability] A sequence of relations $\vec{\boldsymbol{\mathcal{R}}}$ is **simultaneously implicitly elementary definable** on $\mathcal{C}$ iff there is a sentence $\phi(\vec{X})$ in the language $L^2_{\mathcal{C}}$ in which the $\vec{X}$ are the only second-order variables that occur and the only variables that occur free, and each of the X_i in $\vec{X}$ occurs with the appropriate arity for the corresponding relation $\boldsymbol{\mathcal{R}}_i$ in $\vec{\boldsymbol{\mathcal{R}}}$, such that, for each $M \in \mathcal{C}$ and $\boldsymbol{\mathcal{R}}_i$,

$$M \models \phi(\vec{X})[s] \Leftrightarrow \boldsymbol{\mathcal{R}}_i(M) = s(X_i).$$

The essence of implicit definability is that there is exactly one way to expand each $M \in \mathcal{C}$ with an interpretation of X such that the resulting structure satisfies $\phi(X)$, and that is by interpreting X as $\mathcal{R}$. If we take $\mathcal{C}$ to be the class of *all structures* over some signature, then Beth's Definability Theorem (Beth 1953) states that whenever a sequence of relations is implicitly elementary definable on that class then each of those relations are explicitly elementary definable on that class as well. As a result, the distinction between implicit and explicit definability is often ignored. Beth's theorem, however, does not necessarily hold for restricted classes of structures, and, in particular, is known to fail for the class of all finite structures (Gurevich 1984).

It is easy to see, however, that implicit *elementary* definitions add nothing to the expressive power of the second-order language, since every $\mathcal{R}$ that is implicitly elementary definable on $\mathcal{C}$, by $\phi(X)$ say, can be explicitly second-order defined by

$$R(\vec{x}) \leftrightarrow (\exists X)[\phi(X) \wedge X(\vec{x})].$$

Again, we can utilize a predicate R with its denotation fixed to be such an $\mathcal{R}$ without actually leaving $L^2_{\mathcal{C}}$ since each occurrence of $R(\vec{x})$ can be replaced with the right-hand side of its explicit second-order definition.

When we restrict ourselves to $L^2_{K,P}$ this interpretation of implicit first-order definitions as explicit second-order definitions only goes through for monadic predicates, since we have only monadic second-order variables at our disposal. The concomitant restriction to monadic implicit definitions is, in fact, unavoidable. We will show, in Section 5.3.4, that there are binary relations that are implicitly first-order definable but not explicitly second-order definable over trees in $L^2_{K,P}$.

Definition 5 [Implicit Second-order Definability] A second-order relation $\boldsymbol{\mathcal{R}}(M) \subseteq \mathcal{P}(|M|)^n$ is **implicitly second-order definable** on a class of structures $\mathcal{C}$ iff there is a formula $\phi(\boldsymbol{X})$ in the language $L^3_{\mathcal{C}}$ in which $\boldsymbol{X}$ is the only third-order variable that occurs and only $\boldsymbol{X}$ occurs free such that, for all $M \in \mathcal{C}$,

$$M \models \phi(\boldsymbol{X})[s] \Leftrightarrow \boldsymbol{\mathcal{R}}(M) = s(\boldsymbol{X}).$$

The notion of implicit second-order definitions of elementary relations is not useful, since, as we have just seen, any such definition is just an explicit second-order definition. It is meaningful, however, to think in terms of sequences of both elementary and second-order relations that are simultaneously implicitly second-order definable. Again, to be precise, we can think of the elementary relations in such a sequence as relations on singleton subsets.

In general, at least over restricted classes of structures, implicit definitions extend the expressive power of $L^2_{\mathcal{C}}$. We will show, again in Section 5.3.4, that there are implicitly second-order definable second-order relations, even second-order relations with monadic arguments, that are not explicitly second-order definable over trees in $L^2_{K,P}$.

5.1.3 Positive-Inductive Definitions

The notion of positive-inductive definability has figured prominently in the study of descriptive computational complexity (see Chandra and Harel 1982, Gurevich 1988). The properties of elementary positive-inductive definitions (over fixed arbitrary structures) have been explored by Moschovakis (1974). For the most part we follow Moschovakis's exposition, generalizing slightly. The reader is directed there for details.

Suppose $\phi(\vec{x}, X)$ is a $L^2_{\mathcal{C}}$ formula with free variables among $\vec{x}$ and X, no other second-order variables other than X, and in which X only **occurs positively**—within the scope of an even number of negations.

Let $?_\phi$ be the operator mapping $\mathcal{P}(|M|^n) \to \mathcal{P}(|M|^n)$, where n is the arity of X, such that

$$?_\phi(\mathcal{R}) = \{s(\vec{x}) \mid M \models \phi(\vec{x}, X)\,[s], s(X) = \mathcal{R}\}\,.$$

Let

$$?^\kappa_\phi = ?_\phi(\bigcup_{\xi<\kappa}[?^\xi_\phi]),$$

(and thus $?^0_\phi = ?_\phi(\emptyset)$). Since X occurs only positively, $?_\phi$ is **monotone**, that is, if $\mathcal{A} \subseteq \mathcal{B}$ then $?_\phi(\mathcal{A}) \subseteq ?_\phi(\mathcal{B})$. It follows that $?_\phi$ has a least fixed point I_ϕ, which is $?^\kappa_\phi$ for the smallest ordinal κ for which $?(?^\kappa_\phi) = ?^\kappa_\phi$. The cardinality of κ is always less than or equal to that of the universe of M.

Definition 6 [Positive-inductive elementary definability] A relation $\boldsymbol{\mathcal{R}}(M) \subseteq |M|^n$ is **positive-inductively (elementary) definable** on $\mathcal{C}$ iff there is such a formula $\phi(\vec{x}, X)$ for which, for all M in $\boldsymbol{\mathcal{C}}$, $I_\phi = \boldsymbol{\mathcal{R}}(M)$.

While the key characteristic of all definitions is that there is a single relation that satisfies the definition, for positive-inductive definitions there may be many relations $\mathcal{R}$ for which $?_\phi(\mathcal{R}) = \mathcal{R}$, that is, $?_\phi$ may have many fixed points. The essential characteristic of a positive-inductive definition is that there is a unique minimum (with respect to subset) such relation. For elementary definitions, this characteristic is a second-order definable property. Thus, if $\mathcal{R}$ is a relation defined positive-inductively by $\phi(\vec{x}, X)$, it can be explicitly second-order defined with

$$\mathcal{R}(\vec{x}) \leftrightarrow (\forall X)[(\forall \vec{y})[\phi(\vec{y}, X) \rightarrow X(\vec{y})] \rightarrow X(\vec{x})].$$

That is: $\vec{x}$ is in $\mathcal{R}$ iff it is in every relation closed under $?_\phi$—in every fixed-point of $?_\phi$. Here again, then, we can incorporate positive-inductively elementary defined relation symbols into the language without actually leaving $L^2_{\mathcal{C}}$.

Of course this interpretation of positive-inductive elementary definitions as explicit second-order definitions suffers the same limitation when we restrict to $L^2_{K,P}$ as our interpretation of implicit elementary definitions did—it only works for monadic predicates—and we will show, with a minor variation of our argument for the implicit case, that there are binary relations on $\mathcal{N}_\omega$ that are positive-inductively elementary definable but not explicitly second-order definable over trees in $L^2_{K,P}$.

We would like to consider positive-inductive definitions of second-order relations as well. This is actually a straightforward generalization of the elementary case since we can think of definitions of relations on subsets of $|M|$ in terms of definitions of elementary relations on the structure built on the power set of $|M|$.

Suppose $\boldsymbol{X}$ is a second-order variable with monadic arguments. Suppose $\phi(\vec{X}, \boldsymbol{X})$ is a formula with free variables among $\vec{X}$ and $\boldsymbol{X}$, no other third-order variables other than $\boldsymbol{X}$, and in which $\boldsymbol{X}$ only occurs positively.

Let $?_\phi$ map $\mathcal{P}(\mathcal{P}(|M|)^n) \rightarrow \mathcal{P}(\mathcal{P}(|M|)^n)$, where n is the arity of $\boldsymbol{X}$, such that

$$?_\phi(\boldsymbol{\mathcal{R}}) = \left\{ s(\vec{X}) \mid M \models \phi(\vec{X}, \boldsymbol{X})\,[s], s(\boldsymbol{X}) = \boldsymbol{\mathcal{R}} \right\}.$$

Define $?^\kappa_\phi$ as above. Again, $?_\phi$ is monotone and has a least fixed point I_ϕ.

Definition 7 [Positive-inductive second-order definability] A relation $\boldsymbol{\mathcal{R}}(M) \subseteq \mathcal{P}(|M|)^n$ is **positive-inductively second-order definable** on $\mathcal{C}$ iff there is such a formula $\phi(\vec{X}, \boldsymbol{X})$ for which $I_\phi = \boldsymbol{\mathcal{R}}(M)$ for all M in $\mathcal{C}$.

Again, we will be able to show for this class of definitions, as with implicit second-order definitions, that their expressive power is strictly greater than that of explicit second-order definitions—there are second-order relations, even those with monadic arguments, that are positive-inductively second-order definable but not explicitly second-order definable over trees in $L^2_{K,P}$.

5.2 Definability over Trees in $L^2_{K,P}$

From this discussion, it follows that we can expand the language of $L^2_{K,P}$ with explicit elementary or second-order definitions without extending its expressive power. In addition, we can also employ monadic relations that are implicitly or positive-inductively elementary defined. Beyond that, we cannot employ implicit or positive-inductive second-order definitions or implicit or positive-inductive elementary definitions of non-monadic relations without going beyond the descriptive power of $L^2_{K,P}$. In fact, it will be a consequence of our proof of this fact that we cannot employ such definitions without losing solvability.

As a rule, we will be interested in defining classes of labeled trees. In our interpretation, these are relations between subsets of T_ω: the set of all nodes in the tree and the sets of nodes with each of the labels. Thus, we generally will be explicitly defining non-monadic second-order relations. In doing this, we can use monadic first-order relations freely—these are interpreted as monadic predicate variables. Non-monadic relations and second-order relations, on the other hand, must be explicitly defined.

5.2.1 Some Defined Relations

The following relations will be used in the sequel.

We start with a constant for the root:

$$\mathrm{root}(x) \equiv (\forall y)[x \triangleleft^* y]. \tag{1}$$

A subset of T_ω satisfies InclRoot iff it includes the root:

$$\mathrm{InclRoot}(X) \equiv (\exists x)[X(x) \wedge \mathrm{root}(x)]. \tag{2}$$

A weaker condition is rootedness in the sense of having a lower bound wrt domination:

$$\mathrm{Rooted}(X) \equiv (\exists x)(\forall y)[X(x) \wedge (X(y) \rightarrow x \triangleleft^* y)] \tag{3}$$

A subset is connected if domination restricted to the nodes in that set is the reflexive transitive closure of parent similarly restricted:

$$\mathrm{Connected}(X) \equiv (\forall x, y, z)[(X(x) \wedge X(y) \wedge x \triangleleft^* z \wedge z \triangleleft^* y) \rightarrow X(z)]. \tag{4}$$

Partition holds between a sequence of subsets and another subset exactly in case the sequence partitions the other. There is a distinct Partition relation of each arity greater than one.

$$\begin{aligned} &\mathrm{Partition}(\vec{X}, Y) \equiv \\ &\quad (\forall x)[(Y(x) \leftrightarrow \bigvee_{X \in \vec{X}} X(x)) \wedge \bigwedge_{X \in \vec{X}} [X(x) \rightarrow \bigwedge_{Z \in \vec{X} \setminus \{X\}} \neg Z(x)]]. \end{aligned} \tag{5}$$

Path denotes any connected subset that is linearly ordered by domination (we are relaxing the requirement that it have a minimum and maximum member):

$$\begin{aligned} &\mathrm{Path}(X) \equiv \\ &\quad \mathrm{Connected}(X) \wedge (\forall x, y)[(X(x) \wedge X(y) \rightarrow (x \triangleleft^* y \vee y \triangleleft^* x)]. \end{aligned} \tag{6}$$

Branch is a rooted, unbounded path:

$$\begin{aligned} &\mathrm{Branch}(X) \equiv \\ &\quad \mathrm{InclRoot}(X) \wedge \mathrm{Path}(X) \wedge (\forall x)(\exists y)[X(x) \rightarrow (X(y) \wedge x \triangleleft^+ y)]. \end{aligned} \tag{7}$$

Subset:

$$\mathrm{Subset}(X, Y) \equiv (\forall x)[X(x) \rightarrow Y(x)]. \tag{8}$$

Since all subsets of T_ω are well-ordered by the lexicographic order, X is finite iff each of its subsets has an upper-bound wrt $\preceq$:

$$\begin{aligned} &\mathrm{Finite}(X) \equiv \\ &\quad (\forall Y)(\exists x)(\forall y)[\mathrm{Subset}(Y, X) \rightarrow (Y(x) \wedge (Y(y) \rightarrow y \preceq x))]. \end{aligned} \tag{9}$$

5.3 Non-Definability in $L^2_{K,P}$

We have that a relation is definable in $L^2_{K,P}$ iff it is definable in SnS. Thus, definability and non-definability results for SnS apply to $L^2_{K,P}$ as well. This is a reasonably well-studied topic (see Läuchli and Savioz 1987 and Thomas 1990). *Non-definability* results are of particular interest to us as these provide bounds on the relations, and hence, the sets of trees, which we can capture in SnS, and consequently, in $L^2_{K,P}$. By definition, every structure extending $\mathcal{N}_\omega$ with a predicate for a relation definable in SnS will have a decidable theory. One approach, then, to proving that a given relation is not definable in SnS is to show that the theory of $\mathcal{N}_\omega$ extended with a predicate for that relation is not decidable. As with many undecidability results, these ultimately involve reductions from the halting problem. Typically (again in the context of SnS) the reduction is done via the *Origin-Constrained Tiling Problem* which yields non-decidability of the monadic second-order theory of the *grid*. This result is due to Lewis (1979). While we will not repeat his proof here, we will sketch it, adapted to our terminology.

5.3.1 The Origin-Constrained Tiling Problem

A **Tiling System** is a tuple $\boldsymbol{\mathcal{D}} = \langle \mathcal{D}, \mathcal{D}_0, \mathcal{H}, \mathcal{V} \rangle$, where

$$\begin{array}{rcl} \mathcal{D} & & \text{is a finite set} \\ \mathcal{D}_0 & \subseteq & \mathcal{D} \\ \mathcal{H}, \mathcal{V} & \subseteq & \mathcal{D}^2 \end{array}$$

A **Tiling** is a map $\tau : \mathbb{N}^2 \to \mathcal{D}$.

A tiling τ is **accepted** by a tiling system $\boldsymbol{\mathcal{D}}$ iff, for all $\langle x, y \rangle \in \mathbb{N}^2$

$$\begin{array}{rcl} \tau(\langle 0,0 \rangle) & \in & \mathcal{D}_0 \\ \langle \tau(\langle x,y \rangle), \tau(\langle x+1,y \rangle) \rangle & \in & \mathcal{H} \\ \langle \tau(\langle x,y \rangle), \tau(\langle x,y+1 \rangle) \rangle & \in & \mathcal{V} \end{array}$$

The **Origin-Constrained Tiling Problem** is the question of whether, for a given $\boldsymbol{\mathcal{D}}$, there exists a τ such that $\mathcal{D}$ accepts τ.

Lewis shows the origin-constrained tiling problem to be undecidable by reduction from the halting problem. To do this, he chooses

$$\mathcal{D} = \Sigma \cup \{\langle q, \sigma \rangle \mid q \in Q, \sigma \in \Sigma\}$$

where Σ is the set of tape symbols and Q the set of states of a given Turing Machine TM. He then defines $\mathcal{H}$ and $\mathcal{V}$ such that instantaneous descriptions of TM are encoded parallel to the x-axis, their evolution over time is encoded parallel to the y-axis, the initial configuration is required along the $y = 0$ row, and a halting configuration is required to occur, which, consequently, must be the maximum row wrt y. Some refinement is required to account for the fact that the head of TM moves diagonally in this space, etc., but it is reasonably easy to see that the tiling system defined in this way accepts a τ iff that τ encodes a halting computation of TM.

5.3.2 The Monadic Second-Order Theory of the Grid

Let $G = \left\langle \mathbb{N}^2, \mathrm{O}, \mathrm{r}_0, \mathrm{r}_1 \right\rangle$, where

$$\begin{aligned} \mathrm{O} &= \langle 0, 0\rangle \\ \mathrm{r}_0(\langle x, y\rangle) &= \langle x+1, y\rangle \\ \mathrm{r}_1(\langle x, y\rangle) &= \langle x, y+1\rangle \end{aligned}$$

Let $\mathbf{Th}_2(G)$ be the monadic second-order theory of G.

Lemma 1 (Lewis 1979) $\mathbf{Th}_2(G)$ *is undecidable.*

Proof. By reduction from the origin-constrained tiling problem. Given a tiling system $\boldsymbol{\mathcal{D}} = \langle \mathcal{D}, \mathcal{D}_0, \mathcal{H}, \mathcal{V}\rangle$, Let $\vec{D}$ be the set $\mathcal{D}$, in arbitrary order, take as variables. $\boldsymbol{\mathcal{D}}$ can be encoded in the language of G as follows:

$$\begin{aligned} \phi_0^{\mathcal{D}} &= \bigvee_{D\in\mathcal{D}_0} D(\mathrm{O}) \\ \phi_H^{\mathcal{D}} &= (\forall \langle x, y\rangle)[\bigwedge_{D\in\mathcal{D}} [D(\langle x, y\rangle) \to \bigvee_{\langle D, D'\rangle\in\mathcal{H}} [D'(\mathrm{r}_0(\langle x, y\rangle))]]] \\ \phi_V^{\mathcal{D}} &= (\forall \langle x, y\rangle)[\bigwedge_{D\in\mathcal{D}} [D(\langle x, y\rangle) \to \bigvee_{\langle D, D'\rangle\in\mathcal{V}} [D'(\mathrm{r}_1(\langle x, y\rangle))]]] \\ \phi_\tau^{\mathcal{D}} &= (\forall x)[\bigvee_{D\in\mathcal{D}} [D(x)] \wedge \bigwedge_{D\in\mathcal{D}} [D(x) \to \bigwedge_{D'\in\mathcal{S}\setminus D} [\neg D'(x)]]] \end{aligned}$$

Then

$$\begin{aligned} &(\exists\vec{D})[\phi_O^{\mathcal{D}} \wedge \phi_H^{\mathcal{D}} \wedge \phi_V^{\mathcal{D}} \wedge \phi_\tau^{\mathcal{D}}] \in \mathbf{Th}(G) \\ \Leftrightarrow\ &(\exists\tau)[\tau \text{ is accepted by } \mathcal{D}] \end{aligned}$$

In fact, there is such a τ iff

$$G \models (\phi_0^{\mathcal{D}} \wedge \phi_H^{\mathcal{D}} \wedge \phi_V^{\mathcal{D}} \wedge \phi_\tau^{\mathcal{D}})\,[\tau^{-1}],$$

that is, iff the sets of points mapped to $D \in \mathcal{D}$ are satisfying assignments in G for those D. □

Since the subsets of the grid that correspond to halting computations of the Turing Machine are necessarily finite, weak quantification suffices for the proof. Thus, the *weak monadic second-order theory of the grid* is undecidable as well.

Note also that the second-order quantification is needed only to existentially quantify the variables in $\mathcal{D}$. We can define a class of labeled grids in which the variables are explicitly interpreted as predicate constants:

$$G_{\mathcal{D}} = \left\langle \mathbb{N}^2, \mathrm{O}, \mathrm{r}_0, \mathrm{r}_1, D \right\rangle_{D\in\mathcal{D}}.$$

The first-order theory the this class of structures, then, is undecidable by the same argument.

5.3.3 Some Relations Non-Definable in SnS

With this result, we can establish non-definability of a relation in SnS by showing that, using a predicate for that relation, we can define a substructure of $\mathcal{N}_\omega$ that is isomorphic to the grid. As an example, consider a function $s : s(w) \mapsto 0w$ for all $w \in \{0,1\}^*$.[1] We can identify the set 0^*1^* with the grid (with the map $0^i1^j \mapsto \langle i,j\rangle$). This is a definable subset of T_2, by, for instance,

$$G_T(x) \equiv (\exists x_0)(\forall y)[x_0 \triangleleft^* x \wedge y \not\prec x_0 \wedge (x_0 \triangleleft^* y \rightarrow x \not\prec y)].$$

The conjunct $y \not\prec x_0$ insures that x_0 is on the left-most branch of T_2, that is, $x_0 \in 0^*$. The conjunct $x_0 \triangleleft^* y \rightarrow x \not\prec y$ insures that x is on the right-most branch of the subtree rooted at x_0, that is, $x \in 0^*1^*$. The structure $\langle G_T, \epsilon, s, r_1\rangle$, then, is isomorphic to the grid.

A second example is the **equal-level predicate**. This denotes the relation

$$E = \left\{\langle v,w\rangle \mid |v| = |w|,\, v,w \in \{0,1\}^*\right\}.$$

With this, one can define s, since $s(x) = y$ iff

$$\begin{aligned}(\exists x_0, y_0, y_1)(\forall z)[\quad & x_0 \triangleleft^* x \wedge z \not\prec x_0 \wedge (x_0 \triangleleft^* z \rightarrow x \not\prec z) \wedge \\ & x_0 \triangleleft y_0 \wedge y_0 \triangleleft^* y \wedge z \not\prec y_0 \wedge (y_0 \triangleleft^* z \rightarrow y \not\prec z) \wedge \\ & y_1 \triangleleft y \wedge \mathrm{E}(y_1, x)].\end{aligned}$$

Here x_0 and x are as in the definition of 0^*1^*. Similarly for y_0 and y. The node y_0 is the left child of x_0, and consequently, if $x = 0^i1^j$, then $y_0 = 0^{i+1}$.

Suppose $j > 0$. Since the node y_1 is at the same level in the tree as x it is dominated by y_0. Since it is the parent of y, which is on the right-most branch of the subtree rooted at y_0, it must be the case that y_1 is also on the right-most branch of that subtree. Therefore, $y_1 = 0^{i+1}1^{j-1}$ and $y = 0^{i+1}1^j$.

If $j = 0$, on the other hand, then $x_0 = x$. It follows that $y_1 = x$ as well, and $y_0 = y$. Thus, $y = 0^{i+1}$.

5.3.4 The Additional Expressive Power of Implicit and Positive-Inductive Definitions

We can use the non-definability of the equal-level predicate to witness the additional expressive power of implicit and positive-inductive definitions of non-monadic relations and of implicit and positive-inductive second order definitions of monadic relations on the class of trees. We can capture the equal-level predicate with an implicit elementary definition

[1] This example and the next are discussed in Thomas 1990. Proofs, by an alternative method, are given in Läuchli and Savioz 1987.

as the conjunction of the following formulae:

$$(\forall x, y)[x \approx y \rightarrow \mathrm{EL}(x, y)] \tag{10a}$$
$$(\forall x, y)[\mathrm{EL}(y, x) \rightarrow \mathrm{EL}(x, y)] \tag{10b}$$
$$(\forall x, y, z)[(\mathrm{EL}(x, z) \wedge \mathrm{EL}(z, y)) \rightarrow \mathrm{EL}(x, y)] \tag{10c}$$
$$(\forall x, x_0, y, y_0)[(x_0 \triangleleft x \wedge y_0 \triangleleft y \wedge \mathrm{EL}(x_0, y_0)) \rightarrow \mathrm{EL}(x, y)] \tag{10d}$$
$$(\forall x, y)[x \triangleleft^+ y \rightarrow \neg\mathrm{EL}(x, y)] \tag{10e}$$

The first three conditions require EL to be an equivalence relation—reflexive, symmetric, and transitive. Condition 10d insures that EL is true of nodes on the same level. (This is an easy induction on the level of the nodes. Note that the fact that EL is reflexive implies that the root is related to itself by EL.) The final condition insures that nodes that are not on the same level are not related by EL. To see this, assume there are two nodes a and b at different levels that are related by EL. Assume a is the deeper of these. There must be a c, a proper predecessor of a, that is at the same level as b. Since they are equal-level nodes b and c must be related by EL and, thus, by transitivity, a and c must be as well, contradicting condition 10e.

To transform this into a positive-inductive definition we need only drop condition 10e and take the disjunction of the antecedents of the implications of the remaining conditions:

$$x \approx y \quad \vee \tag{11a}$$
$$\mathrm{EL}(y, x) \quad \vee \tag{11b}$$
$$(\exists z)[\mathrm{EL}(x, z) \wedge \mathrm{EL}(z, y)] \quad \vee \tag{11c}$$
$$(\exists x_0, y_0)[x_0 \triangleleft x \wedge y_0 \triangleleft y \wedge \mathrm{EL}(x_0, y_0)] \tag{11d}$$

Again, it is easy to show that ? for this formula preserves the condition 10e.

To capture the equal-level predicate as a monadic second-order relation (a property of sets individuals) we define the class $\boldsymbol{L}$ of **strata** of the tree. A set is in the class $\boldsymbol{L}$ iff it consists of all nodes at some level in the tree. $\mathrm{EL}(x, y)$ is explicitly definable in terms of $\boldsymbol{L}(X)$:

$$\mathrm{EL}(x, y) \equiv (\exists X)[X(x) \wedge X(y) \wedge \boldsymbol{L}(X)]. \tag{12}$$

$\boldsymbol{L}$ can be defined implicitly as follows:

$$(\forall X)[(\forall x)[X(x) \leftrightarrow \mathrm{root}(x)] \rightarrow \boldsymbol{L}(X)] \tag{13a}$$
$$(\forall X)[(\exists Y)[\boldsymbol{L}(Y) \wedge (\forall x)[X(x) \leftrightarrow (\exists y)[Y(y) \wedge y \triangleleft x]]] \rightarrow \boldsymbol{L}(X)] \tag{13b}$$
$$(\forall X)[(\exists x, y)[X(x) \wedge X(y) \wedge x \triangleleft^+ y] \rightarrow \neg\boldsymbol{L}(X)] \tag{13c}$$

Condition 13a states that the set consisting of just the root is in $\boldsymbol{L}$, condition 13b states that the set of children of any set in $\boldsymbol{L}$ is also in $\boldsymbol{L}$, and the final condition states that no set in $\boldsymbol{L}$ contains any individuals related by proper domination.

Again we can convert this to a positive-inductive definition by dropping condition 13c and taking the disjunction of the antecedents of the implications in the first two:

$$(\forall x)[X(x) \leftrightarrow \mathrm{root}(x)] \qquad \vee \tag{14a}$$

$$(\exists Y)[\boldsymbol{L}(Y) \wedge (\forall x)[X(x) \leftrightarrow (\exists y)[Y(y) \wedge y \triangleleft x]]] \tag{14b}$$

5.3.5 The Non-Definability of Subtree Isomorphism

We will derive some additional non-definability results in the next section by other methods, but as a final example of non-definability by reduction from undecidability of the grid, consider the relation $\mathrm{Iso}_{\vec{P}}(x, y)$. This holds iff the subtrees rooted at x and y are isomorphic wrt the predicates in $\vec{P}$. That is, a tree satisfies $\mathrm{Iso}_{\vec{P}}(v, w)$ iff the subtrees rooted at v and w, restricted to the predicates in $\vec{P}$, are isomorphic.

This predicate is significant because it is essentially the difference between the theory of trees as subsets of $\mathcal{N}_\omega$ and the algebraic theory of trees as in Courcelle 1983. If we fix some (finite) set of predicates $\vec{P}$ and define equality extensionally in terms of $\mathrm{Iso}_{\vec{P}}$, that is, take all pairs of individuals related by $\mathrm{Iso}_{\vec{P}}(x, y)$ to be equal, then the theory of the resulting class of structures is just the algebraic theory of trees over $\vec{P}$. As this is not a decidable theory, it should come as no surprise that $\mathrm{Iso}_{\vec{P}}$ is not definable in $L^2_{K,P}$.

We show this by showing that, using $\mathrm{Iso}_{\vec{P}}$, we can capture an undecidable fragment of the monadic second-order theory of the grid using the property (of trees, parameterized by $\vec{P}$):

$$\begin{aligned}\Phi_I(\vec{P}) \;\equiv\; (\forall x, x_0, x_1, y_0, y_1)[&\\ &(x \triangleleft x_0 \wedge x \triangleleft y_0 \wedge x_0 \prec y_0 \wedge\\ &x_0 \triangleleft x_1 \wedge y_0 \triangleleft y_1 \wedge\\ &(\forall z)[x_1 \not\prec z \vee z \not\prec y_1]) \quad \rightarrow \quad \mathrm{Iso}_{\vec{P}}(x_1, y_1)\,].\end{aligned} \tag{15}$$

The nodes x_0 and y_0 are the left and right children of x, respectively $x0$ and $x1$. Since no nodes fall between x_1 and y_1 with respect to left-of, these must be, in turn, the right child of x_0 and left child of y_1. This, then, simply requires the subtrees at $x01$ and $x10$ to be isomorphic with respect to the assignment of the predicates $\vec{P}$.

Theorem 2 $\mathrm{Iso}_{\vec{P}}(x, y)$ *is not definable in* SnS.

Suppose, for contradiction, that $\mathrm{Iso}_{\vec{P}}(x, y)$ is a formula in the language of SnS that is true iff the subtrees rooted at the nodes assigned to x and y are isomorphic wrt the sets assigned to $\vec{P}$. Then $\Phi_I(\vec{P})$ is a formula in the language of SnS. We show that there are sentences in which $\Phi_I(\vec{P})$ occurs as a subformula that would then be in S2S, but for which satisfiability is non-decidable, contradicting the decidability of S2S.

The underlying idea of the proof is that we can interpret paths from the root in $\mathcal{N}_2$ as paths from the origin in $\mathbb{N}^2$ that follow the same sequence of successors. This defines a many-to-one map from T_2 onto $\mathbb{N}^2$. The formula $\Phi_I(\vec{P})$ is sufficient to insure that all nodes in T_2 that map to the same point in $\mathbb{N}^2$ are labeled identically by the interpretations of $\vec{P}$. It will follow, then, that

$$(\exists \vec{D})[\phi^{\mathcal{D}}_O \wedge \phi^{\mathcal{D}}_H \wedge \phi^{\mathcal{D}}_V \wedge \phi^{\mathcal{D}}_\tau \wedge \Phi_I(\vec{D})]$$

will be in S2S iff

$$(\exists \vec{D})[\phi^{\mathcal{D}}_O \wedge \phi^{\mathcal{D}}_H \wedge \phi^{\mathcal{D}}_V \wedge \phi^{\mathcal{D}}_\tau]$$

is in $\mathbf{Th}_2(G)$. As this is the formula from Section 5.3.2 that captures the tiling system $\boldsymbol{\mathcal{D}}$, its satisfiability is non-decidable.

Let $M = [X_{\mathcal{U}} \mapsto T_2, P_i \mapsto P_i^M]_{P_i \in \vec{P}}$ be a labeled tree in $\mathcal{N}_2$ that satisfies $\Phi_I(\vec{P})$. Such an M exists—the tree in which every P_i^M is either T_2 or the empty set, for instance.

For the sake of simplicity, if $v, w \in T_2$ we will say $\text{Iso}_{\vec{P}}(v, w)$ holds to indicate that $M \models \text{Iso}_{\vec{P}}(x, y)[x \mapsto v, y \mapsto w]$.

Suppose $v \in T_2$. (Recall that $T_2 = \{0,1\}^*$.) We will denote the number of '0's in v by $|v|_0$, and the number of '1's by $|v|_1$. Let

$$\mathcal{E}_G = \{\langle v, w\rangle \mid |v|_0 = |w|_0 \text{ and } |v|_1 = |w|_1\} \qquad (16)$$

Then $\langle v, w\rangle \in \mathcal{E}_G$ iff v and w have the same end point when interpreted as paths from the origin in $\mathbb{N}^2$. Further, $\langle v, w\rangle \in \mathcal{E}_G$ iff v is a permutation of w.

Lemma 3 *If $M \models \Phi_I(\vec{P})$, then $\langle v, w\rangle \in \mathcal{E}_G$ implies $\text{Iso}_{\vec{P}}(v, w)$.*

Proof. We must show that whenever v is a permutation of w, then the subtrees at v and w are isomorphic wrt the interpretations of $\vec{P}$ in M. We do this by induction on $|v|$, the length of v (and of w, as well).

Suppose $|v| = 0$. Then $v = w$ and $\text{Iso}_{\vec{P}}(v, w)$ holds trivially.

Suppose, then, that $|v| \geq 1$ and the result holds for all pairs of nodes at depths in the tree less than $|v|$.

Suppose, further, that $v = v'i$ and $w = w'i$ for some $i \in \{0,1\}$. Then, by the induction hypothesis, $\text{Iso}_{\vec{P}}(v', w')$. It follows that $\text{Iso}_{\vec{P}}(v, w)$ holds.

Suppose, alternatively, that v and w differ in their last element. Without loss of generality, let v end in '0' while w ends in '1'. Since v is a permutation of w, it must be the case that $v = v'1v''0$ and $w = w'0w''1$

for some $v', v'', w', w'' \in \{0,1\}^*$. Then

$$\langle v'1v'', v'v''1\rangle, \langle w'0w'', w'w''0\rangle, \langle v'v'', w'w''\rangle \in \mathcal{E}_G$$

and, since these are all shorter than $|v|$, these are all related by $\mathrm{Iso}_{\vec{P}}$.

By 15, $\mathrm{Iso}_{\vec{P}}(v'v''10, v'v''01)$. Then

$$\begin{array}{ll} \mathrm{Iso}_{\vec{P}}(v'v''10, w'w''01) & \text{by isomorphism of } v'v'' \text{ and } w'w'' \\ \mathrm{Iso}_{\vec{P}}(v'1v''0, w'w''01) & \text{by isomorphism of } v'1v'' \text{ and } v'v''1 \\ \mathrm{Iso}_{\vec{P}}(v'1v''0, w'0w''1) & \text{by isomorphism of } w'0w'' \text{ and } w'w''0 \end{array}$$

Which is just $\mathrm{Iso}_{\vec{P}}(v, w)$. □

Thus, the subtrees rooted at pairs of points in T_2 that correspond to the same point in $\mathbb{N}^2$ must be labeled identically with the $\vec{P}$ by M if $M \models \Phi_I(G)$. This lets us take the quotient of M wrt $\mathcal{E}_G$.

Let $[v]_G = \{w \mid \langle v, w\rangle \in \mathcal{E}_G\}$.

Let $\mathcal{N}_G = \left\langle T_2/_{\mathcal{E}_G}, \epsilon, \mathrm{r}_0^G, \mathrm{r}_1^G \right\rangle$, where

$$\begin{array}{rcl} T_2/_{\mathcal{E}_G} & = & \{[v]_G \mid v \in T_2\} \\ \mathrm{r}_0^G([v]_G) & = & [v0]_G \\ \mathrm{r}_1^G([v]_G) & = & [v1]_G \end{array}$$

r_i^G is well-defined, since $|v|_0 = |w|_0$ and $|v|_1 = |w|_1$ imply that $|vi|_0 = |wi|_0$ and $|vi|_1 = |wi|_1$.

Let $P_i^G = \left\{[v]_G \mid v \in P_i^M\right\}$.

Since $w \in [v]_G$ implies $w \in P_i^M \Leftrightarrow v \in P_i^M$ (by Lemma 3), $P_i^M = \left\{v \mid [v]_G \in P_i^G\right\}$ as well.

Let $M_G = [P_i \mapsto P_i^G]_{P_i \in \vec{P}}$. M_G is an assignment for $\mathcal{N}_G$. Just as we say that M satisfies a formula iff it is a satisfying assignment for the variables in that formula for $\mathcal{N}_2$, we will say that M_G satisfies a formula iff it is a satisfying assignment for the variables in that formula for $\mathcal{N}_G$.

For the class of formulae relevant to the tiling problem—the class of formulae involving only $\boldsymbol{r}$, the successor functions, the logical connectives and the some set $\vec{P}$ of monadic predicate variables—this quotient preserves satisfiability in the sense that M is a satisfying assignment in $\mathcal{N}_2$ iff M_G is a satisfying assignment in $\mathcal{N}_G$. (This is still under the assumption that M satisfies $\Phi_I(\vec{P})$.)

Lemma 4 *For $\phi(\vec{P})$ in the class just defined,*

$$M \models \phi(\vec{P}) \wedge \Phi_I(\vec{P}) \textit{ iff } M_G \models \phi(\vec{P}).$$

Proof. The proof is by structural induction. We give only the base case and the induction step for one successor function.

$$M \models P_i(x)[x \mapsto v, P_i \mapsto P_i^M]$$

$$
\begin{aligned}
&\Leftrightarrow \quad v \in P_i^M \\
&\Leftrightarrow \quad [v]_G \in P_i^G \\
&\Leftrightarrow \quad M_G \models P_i(x)[x \mapsto [v]_G, P_i \mapsto P_i^G] \\
M &\models P_i(\mathrm{r}_0(x))[x \mapsto v, P_i \mapsto P_i^M] \\
&\Leftrightarrow \quad v0 \in P_i^M \\
&\Leftrightarrow \quad [v0]_G \in P_i^G \\
&\Leftrightarrow \quad M_G \models P_i(\mathrm{r}_0(x))[x \mapsto [v]_G, P_i \mapsto P_i^G]
\end{aligned}
$$

To complete the proof of the $\Leftarrow$ direction, we need to show that the M induced by M_G (by $P_i^M = \{v \mid [v]_G \in P_i^G\}$) satisfies $\Phi_I(\vec{P})$ as well as $\phi(\vec{P})$. This is an easy consequence of the fact that $\mathrm{Iso}_{\vec{P}}(v, w)$ holds in such an M for every pair $\langle v, w\rangle \in \mathcal{E}_G$. $\square$

Note that the restriction to the class $\phi(\vec{P})$ is required, since none of $\approx$, $\triangleleft^*$, or $\leq$ are preserved by this quotient.

To complete the proof of the theorem, we need only to show that M_G models a $\phi(\vec{P})$ sentence iff G models it. This follows nearly immediately from the following lemma, which asserts that M_G is isomorphic to G.

Lemma 5 $M_G \cong G$

Proof. The isomorphism is witnessed by the map

$$h : [v]_G \mapsto \langle |v|_0 \, , |v|_1 \rangle \, .$$

Its trivial to verify that this is a bijective homomorphism from M_G to G. $\square$

It follows that M_G and G are elementary equivalent. Further,

Corollary 6 $M_G \models (\exists \vec{P})[\phi(\vec{P})] \Leftrightarrow G \models (\exists \vec{P})[\phi(\vec{P})]$

as witnessed by the map between subsets induced by h

$$\hat{h} : \{[v]_G \mid v \in T_2\} \mapsto \{\langle |v|_0 \, , |v|_1 \rangle \mid v \in T_2\} \, .$$

We now have all we need to prove the theorem.

Proof. **(Theorem 2)** Given a tiling system $\mathcal{D}$, we have:

$$
\begin{array}{lll}
(\exists\tau)[\tau \text{ accepted by } \mathcal{D}] & & \\
\quad\Leftrightarrow\quad (\exists\vec{D})[\phi^{\mathcal{D}}_O \wedge \phi^{\mathcal{D}}_H \wedge \phi^{\mathcal{D}}_V \wedge \phi^{\mathcal{D}}_\tau] \in \mathbf{Th}_2(G) & & \text{from proof of Lemma 1} \\
\quad\Leftrightarrow\quad G \models (\exists\vec{D})[\phi^{\mathcal{D}}_O \wedge \phi^{\mathcal{D}}_H \wedge \phi^{\mathcal{D}}_V \wedge \phi^{\mathcal{D}}_\tau] & & \text{by def. of } \mathbf{Th}_2 \\
\quad\Leftrightarrow\quad \mathcal{N}_G \models \phi^{\mathcal{D}}_O \wedge \phi^{\mathcal{D}}_H \wedge \phi^{\mathcal{D}}_V \wedge \phi^{\mathcal{D}}_\tau[D_i \mapsto D^G_i] & & \text{by Cor. 6} \\
\quad\Leftrightarrow\quad \mathcal{N}_2 \models \phi^{\mathcal{D}}_O \wedge \phi^{\mathcal{D}}_H \wedge \phi^{\mathcal{D}}_V \wedge \phi^{\mathcal{D}}_\tau \wedge \Phi_I(\vec{D})[D_i \mapsto D^M_i] & & \text{by Lemma 4} \\
\quad\Leftrightarrow\quad \neg(\exists\vec{D})[\phi^{\mathcal{D}}_O \wedge \phi^{\mathcal{D}}_H \wedge \phi^{\mathcal{D}}_V \wedge \phi^{\mathcal{D}}_\tau \wedge \Phi_I(\vec{D})] \notin \text{S2S} & & \text{by def. of S2S}
\end{array}
$$

□

Both the non-definability in $L^2_{K,P}$ of the equal-level predicate and that of $\mathrm{Iso}_{\vec{P}}$ will be useful to us in the next chapter.

5.4 The Class of Sets of Trees Definable in $L^2_{K,P}$

A set of trees is definable in $L^2_{K,P}$ iff membership in the language is a property of subsets of T_ω (a second-order property) that is (explicitly) definable in SωS. Clearly, this class of sets is closed under Boolean operations. Further, since SωS is decidable and emptiness for any set in the class corresponds to satisfiability of $(\exists X)[\phi(X)]$ where $\phi(X)$ is the formula defining the set, emptiness is decidable for every set in the class. This is sufficient to get a number of non-definability results.

Decidability of emptiness immediately implies that the set of derivation trees for an arbitrary context-sensitive grammar[2] is not definable in $L^2_{K,P}$, since this would imply decidability of emptiness of the string language defined by the grammar. Decidability of emptiness along with closure under Boolean operations gives decidability of emptiness of intersection. This gives non-definability of the context-free tree languages of Rounds 1970. It also gives non-definability of the set of trees generated by Tree-Adjoining Grammars (TAGs, Joshi et al. 1975, Joshi 1987, Vijay-Shanker 1987) in which nodes are allowed to be re-labeled during adjunction. In both these cases, non-decidability of emptiness of intersection comes from the ability to define, for any context-free string language, a set of linear trees in which the labeling of the (only) branch of each tree in the set is a word in the string language (and v.v.). Thus, Post's correspondence problem (PCP) reduces, via emptiness of intersection of context-free string languages, to emptiness of intersection of these tree sets.[3]

[2] For the notion of derivation tree to meaningful, of course, the CSG must be in canonical form—each rule must rewrite only a single non-terminal.

[3] We can show that both the sets of CSG-generated trees and the TAG tree sets are positive-inductively second-order definable on $\mathcal{N}_\omega$. In addition, the TAG tree sets are

The ability to capture context-free string languages in this way does not appear to extend to *pure* TAGs—those in which nodes may not be re-labeled, or at least in which labels may not be removed from nodes. It turns out, however, to be reasonably easy to reduce PCP directly to emptiness of the intersection of three pure TAG sets.

As an initial example of a class of tree sets that *are* definable in $L^2_{K,P}$ we have the class of sets of derivation trees generated by context-free (string) grammars. Sets of trees in this class are referred to as **local sets**.

Lemma 7 *The set of derivation trees generated by an arbitrary context-free grammar is definable in* $L^2_{K,P}$.

Proof. Suppose G is a CFG with start symbol S, non-terminals $\boldsymbol{N}$, and terminals $\boldsymbol{A}$. Let $\vec{X}$ be the set $\{X_N \mid N \in \boldsymbol{N}\} \cup \{X_a \mid a \in \boldsymbol{A}\} \cup \{X_\epsilon\}$, in some order.

For each $z \in \boldsymbol{A} \cup \{\epsilon\}$ let

$$\phi_z(\vec{X}, X_G) \equiv (\forall x, y)[(X_z(x) \wedge x \triangleleft^+ y) \rightarrow \neg X_G(y)].$$

For each $N \in \boldsymbol{N}$ where the set of all right-hand sides of productions for N is

$$\{Z_{0,0} \ldots Z_{0,l_0}, \ \ldots, \ Z_{n,0} \ldots Z_{n,l_n}\},$$

where the $Z_{i,j} \in \boldsymbol{N} \cup \boldsymbol{A} \cup \{\epsilon\}$, let

$$\begin{aligned} \phi_N(\vec{X}, X_G) \quad \equiv \quad & (\forall x, y)[X_N(x) \rightarrow \\ & \qquad \bigvee_{i \leq n} [\bigwedge_{j \leq l_i} [X_{Z_{i,j}}(xr_j)] \wedge \\ & \qquad\quad ((x \triangleleft y \wedge xr_{l_i} \prec y) \rightarrow \neg X_G(y))]. \end{aligned}$$

Then let

$$\begin{aligned} \phi_G(\vec{X}, X_G) \quad \equiv \quad & \mathrm{Finite}(X_G) \wedge \mathrm{Partition}(\vec{X}, X_G) \wedge \\ & \bigwedge_{z \in \boldsymbol{A} \cup \{\epsilon\}} \phi_z(\vec{X}, X_G) \wedge \\ & \bigwedge_{N \in \boldsymbol{N}} \phi_N(\vec{X}) \wedge \\ & (\exists x)[X_G(x) \wedge \mathrm{root}(x) \wedge X_S(x)]. \end{aligned}$$

The ϕ_z insure that no terminals are expanded, the ϕ_N insure that each non-terminal is expanded according to some rule in the grammar,

positive-inductively elementary definable on $\mathcal{N}_\omega$ with non-monadic predicates. This confirms that the languages of $L^2_{K,P}$ plus positive-inductive second-order definitions on $\mathcal{N}_\omega$ and that of $L^2_{K,P}$ plus non-monadic positive-inductive elementary definitions on $\mathcal{N}_\omega$ are strictly stronger that that of $L^2_{K,P}$ alone.

Finite and Partition insure that X_G is finite and completely labeled by the symbols of G, and the final conjunct insures that the root is labeled with the start symbol. □

The definability of local sets raises the question of why decidability of emptiness of intersection is not violated. The answer is that emptiness of the intersection of the *string* languages generated by CFGs is undecidable. Here we have defined the *tree* set generated by the CFGs. The undecidability of emptiness of intersection follows from a reduction from PCP and that reduction depends crucially on the existence of strings in the intersection with non-isomorphic derivation trees. The existence of a tree in the intersection of the two languages, of course, implies that the same derivation tree is generated by both grammars. From this observation we get another non-definability result:

Corollary 8 *The relation* $\mathrm{YieldsEq}_{\vec{P}}(X, Y)$*, which holds between finite trees* X *and* Y *iff their yields, i.e., the set of maximal nodes ordered by left-of, are labeled identically by the* $\vec{P}$*, is not definable in* $L^2_{K,P}$.

5.4.1 Characterizing the Local Sets

This last result raises an interesting question. Vijay-Shanker, Weir, and Joshi (1987) discuss a hierarchy of families of languages (originally due to Weir 1987) in which the CFLs and the TALs form the first two levels. Yet here we have the CFG-generated tree sets definable in the second-order existential fragment of $L^2_{K,P}$ while the TAG tree sets are not definable at any level in $L^2_{K,P}$. One may ask what, if any, sets of trees falling between these two are definable in $L^2_{K,P}$. It turns out, via results originally due to Thatcher (1967) and Doner (1970) (in a slightly different form), that every set of finite trees that is definable in $L^2_{K,P}$ is a projection of a set of trees generated by a finite set of CFGs. Thus, modulo the projection and the finiteness restriction, $L^2_{K,P}$ characterizes the CFG-generated tree sets. Doner's work provides the connection to SωS, Thatcher's work provides the basic characterization of local sets in terms of **recognizable sets**—those accepted by automata over *finite* trees. Our proof is essentially a variation and slight extension of that of Thatcher.

Definition 8 [(Mathematical) projection] A **projection onto the** i^{th} **co-ordinate** is a mapping taking tuples to their i^{th} co-ordinate. If Σ is a set of tuples and T is a Σ-valued tree, then the i^{th} projection of T is that tree in which each node is labeled with the i^{th} projection of its label in T. The i^{th} projection of a set of tuples (respectively, a set of tuple-labeled trees) is the set of i^{th} projections of the tuples in the set (respectively, the trees in the set).

We generalize this notion to include projections that take tuples into tuples formed from some subsequence of their co-ordinates, that is, projections that delete certain co-ordinates. We interpret trees decorated with labels from some set $\{X_1, \ldots, X_n\}$ as an assignment of sets of nodes to the X_i. Equivalently (following Rabin), we can interpret these as Σ-labeled trees for $\Sigma = \{0,1\}^n$ in which the i^{th} projection of the tuple labeling a node is 1 just in case that node is included in the set assigned to X_i. A projection of a set of labeled trees, then, is a mapping that suppresses some of the labels.

Lemma 9 *Every set of finite trees with bounded branching that is definable in $L^2_{K,P}$ is the projection of a set of trees generated by a finite set of Context-Free (string) Grammars.*

Theorem 10 *A set of finite trees with bounded branching is local (modulo projection) iff it is definable in $L^2_{K,P}$.*[4]

Proof. **(Lemma 9)** From the discussion above, we have that a set of trees is definable in $L^2_{K,P}$ if every tree in the set is the satisfying assignment for X, with labels that are the satisfying assignments for $\vec{X}$, in a formula $\phi(X, \vec{X})$ in $\mathcal{N}_\omega$. We will assume the trees are labeled subsets of T_2. The generalization to T_n for any finite n is straightforward. By Rabin's theorem there is an automaton $\mathcal{A}_\phi$ on infinite $\{0,1\}^n$-valued trees, where $n = \left|\vec{X}\right| + 1$, such that every tree in the language accepted by the automaton is labeled with tuples built from the characteristic functions of a satisfying assignment for $\phi(X, \vec{X})$. Assume, without loss of generality, that if T is such a tree, then $\boldsymbol{\pi}_1 \circ T$ is the characteristic function of the assignment for X. Thus, the subset of T in which $\boldsymbol{\pi}_1 \circ T$ has the value 1 is isomorphic to a tree in the original set. Call this subset T_X.

Since the trees are necessarily rooted and connected and the behavior of the automaton at any node is unaffected by the labeling of nodes that do not dominate it, the labeling of the nodes in T_X are unaffected by the labeling of the nodes in its complement. Thus, wlog, we may assume for all $w \in T_2$ that $\boldsymbol{\pi}_1 \circ T(w) = 0$ iff $T(w) = 0^n$. Then, since all subtrees of T that are rooted at a node w for which $T(w) = 0^n$ are identical, we may assume that every accepting run of $\mathcal{A}_\phi$ labels them with some state

[4] Bob Frank has pointed out that this result gives us another approach to Corollary 8 of the preceding section, since using $\text{YieldsEq}_{\vec{P}}$ one could generated the language $\{ww \mid w \in \{a,b\}^*\}$. The approach through reduction from PCP, though, is somewhat stronger since it implies that the theory one gets by extending SnS with $\text{YieldsEq}_{\vec{P}}$ is non-decidable, while the approach through reduction from non-context-freeness of ww only implies that it is not equivalent to SnS.

q_f that does not otherwise occur in the run, i.e.,

$$\Delta(q, 0^n) = \begin{cases} \{\langle q_f, q_f \rangle\} & \text{if } q = q_f, \\ \uparrow & \text{otherwise }. \end{cases}$$

Thus, the set of accepting subsets of states for $\mathcal{A}_\phi$ is just $\{\{q_f\}\}$.

Suppose $\mathcal{A}_\phi = \langle Q, q_0, \Delta, \{\{q_f\}\}\rangle$ is such an automaton for ϕ.

Let $G_{\mathcal{A}_\phi}$ be the set of productions:

$$G_{\mathcal{A}_\phi} \stackrel{\text{def}}{=} \{\langle q, \sigma \rangle \longrightarrow \langle q_l, \sigma_l \rangle \, \langle q_r, \sigma_r \rangle \\ \mid \langle q_l, q_r \rangle \in \Delta(q, \sigma) \text{ and } \Delta(q_l, \sigma_l){\downarrow} \text{ and } \Delta(q_r, \sigma_r){\downarrow}\}.$$

Let $S_0 \stackrel{\text{def}}{=} \big\{\langle q_0, \sigma \rangle \mid \langle q_0, \sigma \rangle \longrightarrow lr \in G_{\mathcal{A}_\phi}$ for some l and $r\big\}$. S_0 is, of course, finite.

$G_{\mathcal{A}_\phi}$ can be converted to a finite set of CFGs by the following procedure:

1. Delete the production $\langle q_f, 0^n \rangle \longrightarrow \langle q_f, 0^n \rangle \, \langle q_f, 0^n \rangle$.
2. Delete all occurrences of $\langle q_f, 0^n \rangle$ from all other productions.
3. For each production $\langle q, \sigma \rangle \longrightarrow$ with an empty rhs that results:
   a. delete the production,
   b. add $\langle q, \sigma \rangle'$ to a set of terminal symbols,
   c. add variants of every production in which $\langle q, \sigma \rangle$ occurs in the rhs with $\langle q, \sigma \rangle'$ replacing one or more of those occurrences.
4. For each pair $\langle q_0, \sigma \rangle$ in S_0 define a CFG with $\langle q_0, \sigma \rangle$ as the start symbol, the remaining productions, and the set of terminal symbols collected in the previous step.

Suppose r is an accepting run of $\mathcal{A}_\phi$ on T. We will show that T_X is a projection of a tree generated by one of the CFG constructed above. For all $w \in T_2$ we have $\Delta(r(w), T(w)){\downarrow}$ and $\langle r(w0), r(w1) \rangle \in \Delta(r(w), T(w))$, since r is a run. Thus

$$\langle r(w0), T(w) \rangle \longrightarrow \langle r(w0), T(w0) \rangle \, \langle r(w1), T(w1) \rangle \in G_{\mathcal{A}_\phi}.$$

Let T' be the tree generated by the CFG constructed above with start state $\langle q_0, T(\epsilon) \rangle$ using the productions selected by r and T. Clearly, the right projection of T' is T_X.

Conversely, suppose T' is generated by one of the CFG constructed above. Let T be the right projection of T', extended to a total function on T_2 with $w \mapsto 0^n$ for every $w \in T_2$ that does not occur in T'. Let r be the left projection of T', extended similarly with $w \mapsto q_f$. Then r is an accepting run of $\mathcal{A}_\phi$ on T. □

5.4.2 The Linguistic Significance of (Mathematical) Projection

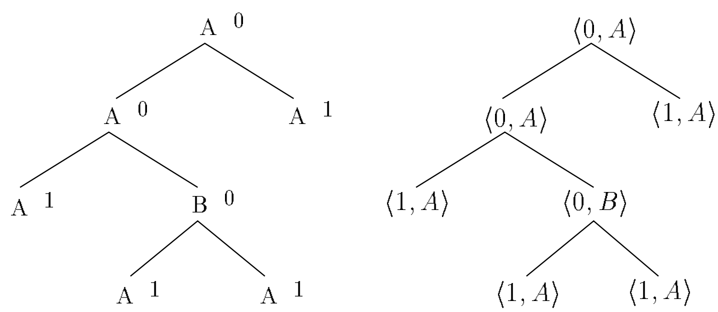

FIGURE 2 Separating Local from Recognizable Sets

The fact that Theorem 10 only holds modulo a projection is somewhat unsatisfying, but the projection is, in fact, necessary—the local sets are a proper subset of the recognizable sets. One set of trees that separates these classes is the set of all finite binary $\{A, B\}$-labeled trees in which exactly one node is labeled B. The tree of Figure 2 is one such tree. This set of trees is recognizable by an automaton that distinguishes nodes that dominate a B from those that do not by the state it assigns to them (0 and 1, respectively, for instance, as in the left-hand tree of the figure).[5] That the set is not local is obvious—if an A can ever expand to AB then any A can expand to AB. The construction of the proof of Lemma 9, in essence, builds a CFG that generates trees labeled with pairs consisting of the state assigned to a node by some successful run and the original label of that node (as in the right-hand tree). The generated grammar includes (among others) productions like

$$\begin{aligned}
\langle 0, A\rangle &\longrightarrow \langle 0, A\rangle\,\langle 1, A\rangle \mid \langle 1, A\rangle\,\langle 0, A\rangle \mid \langle 0, B\rangle\,\langle 1, A\rangle \mid \langle 1, A\rangle\,\langle 0, B\rangle \\
\langle 1, A\rangle &\longrightarrow \langle 1, A\rangle\,\langle 1, A\rangle \\
\langle 0, B\rangle &\longrightarrow \langle 1, A\rangle\,\langle 1, A\rangle
\end{aligned}$$

and so on.

In linguistic terms, the CFG refines the categories of the original set of trees on the basis of a limited amount of context, specifically the context encoded by the state of the automaton; a $\langle 0, A\rangle$ is an A with an embedded B, while a $\langle 1, A\rangle$ is an A that includes no such B. This is essentially the same mechanism as the slashed categories of GPSG.[6] While one might challenge the legitimacy of, for instance, a VP with an NP

[5] We will ignore complications having to do with finiteness, binary branching, etc.

[6] In fact the restrictions on the mechanism are the same as the fundamental limitation on category-valued features in GPSG—we are limited to distinguishing finitely many categories in all.

gap as a category distinct from the class of all VPs, the idea that categories are subdivided on the basis of various additional features is hardly controversial. The refinement introduced by the construction posits features sufficient to make the language context-free. The projection simply ignores those features.

6

Conclusion of Part I

The central result of this part, Theorem 10, is a kind of *descriptive complexity* result for language-theoretic complexity. It says that definability in $L^2_{K,P}$ characterizes (modulo projection) the local sets—the sets of finite trees generated by CFGs. This gives us a powerful tool for investigating the "context-freeness" of grammar formalisms, particularly those that are couched in logical terms possibly far removed from traditional phrase structure rules.

Although the use of $L^2_{K,P}$ is novel, the underlying idea is not. Peters and Ritchie (1969) show that the class of trees **analyzed** by context-sensitive grammars is generated by CFGs. Trees analyzed by a CSG are those trees that are accepted when the grammar is used to verify, for each node that has been expanded by a rule, that the appropriate context can be found at some (*any*) level in the tree. Joshi and Levy (1977) extended this approach to **local constraints**—specifications that include context along branches as well as across the breadth of the tree. These results are related to ours by the fact that they were proved by demonstrating (finite) tree automata that accepted the languages in question.

The result itself is implicit in Doner 1970 and an nearly immediate consequence of the combination of Thatcher and Wright 1968 and Thatcher 1967. Both Doner 1970 and Thatcher and Wright 1968 (independently) address the decidability of weak SnS. Doner 1970 proves the claim that a string language is definable as the yield of a set of finite trees definable in wSnS iff it is a CFL by proving, in essence, Theorem 10. Thatcher and Wright 1968 give an automata-theoretic proof of decidability of wSnS, but don't make the connection to local sets. Thatcher 1967, however, contemporaneously shows the equivalence of local sets and projections of recognizable sets.

The result (in terms of wSnS) is itself a nice extension of an earlier result of Büchi's (1960) (and, again independently, Elgot 1961) establishing the equivalence of regular languages and strings definable in wS1S. Thus, in lifting from a single successor function to multiple successor functions we go from strings to trees, from regular sets to recognizable sets, and (roughly) from regular grammars to context-free grammars.

More recently, Kracht has been pursuing a program that is remarkably similar to ours. In Kracht 1995 he gives constructions for compiling sets of principles stated in the **orientation language**—a particular fragment of dynamic logic—into GPSG-style grammars. The sets of finite trees definable in this fragment of dynamic logic, therefore, are the local sets in an appropriately generalized form.[1] The goal of Kracht's program, as with ours, is to provide a means of comparing, in formal language-theoretic terms, grammar formalisms based on systems of constraints on trees.

The results we have presented here, then, have a place among a number of collections of similar results that differ mostly in the languages they use to express constraints. Although exact equivalence will not necessarily obtain in all cases, it is possible to translate systems of constraints on finite trees stated in $L^2_{K,P}$ into Joshi and Levy's *local constraints*, into wSnS, or into Kracht's *orientation language* and *vice versa*. While the notion of the "naturalness" of logical languages is a matter of personal taste, $L^2_{K,P}$ is a formal language of considerable clarity, and one that is quite close to the languages in which constraints on trees are typically stated. Even though the very restrictions that are responsible for the language's capacity to provide language-theoretic complexity results necessitate a certain amount of cleverness in encoding some linguistic principles, for the most part the difficulty of comprehending the formalization is not tremendously greater than the difficulty of comprehending the original principles.

In the second part of this book, we apply this result to the question of the complexity of the language defined by a Government and Binding Theory account of English, and raise the possibility of using similar results to establish bounds on the generative capacity of GB.

[1]Kracht gives conditions under which such a set of trees is actually a local set in terms of the *memory* of the grammar, a notion he defines. The distinction between local and recognizable sets is closely related to the distinction between *rational* (memory 0) grammars and grammars with non-zero memory.

Part II

The Generative Capacity of GB Theories

7

Introduction to Part II

The result characterizing the local sets by definability in $L^2_{K,P}$ is a kind of descriptive complexity result for language-theoretic complexity—a logical characterization of a language complexity class. Modulo projection, any set of trees we can define within $L^2_{K,P}$ can be generated by a context-free grammar. Conversely, any set of trees that is provably non-definable in $L^2_{K,P}$ is strictly non-context-free. Definability in $L^2_{K,P}$, then, coupled with the numerous non-definability results for SnS, provides a new set of tools for determining context-freeness.

One particularly inviting domain for application of this technique is *Government and Binding Theory* (GB). The principles employed in GB generally have reasonably direct interpretations in formal logic; so much so that a number of the principle-based parsers—parsers for GB-style grammars—are defined directly in a formal logic and implemented in PROLOG (Johnson 1989, Fong 1991, Stabler, Jr. 1992, Cornell 1992). Thus, it is natural to ask which of these principles, or rather which collections of these principles, are definable in $L^2_{K,P}$. The answer to this question would provide a step towards establishing formal bounds on the principles, and consequently, bounds on the generative capacity of GB.

This is the program we undertake here. Our primary result is that a set of principles sufficient to define substantially all of common English syntax is definable in $L^2_{K,P}$. The only non-standard aspect of the set of principles we capture is that we assume a constant bound on the number of overlapping chains that can occur. We show that this is a property of analyses that have been proposed for a broad range of movement in English. Thus, we claim that the set of phrase markers licensed by a reasonably mainstream GB grammar for English is strongly context-free. If one accepts that the language this theory licenses is, in fact, English, then we have a consequential claim that English is a context-free language. We do not, on the other hand, make any such claims about the class of natural languages in general. On the contrary, we explore some specific structures that have been offered as evidence of the non-context-freeness of natural language as a class, and show how our formalization fails to account for these.

The second result we offer here is a non-definability result. We show that the mechanism of *free-indexation*, which occurs widely in GB theories, is beyond the power of $L^2_{K,P}$. Consequently, theories that necessarily employ it license non-context-free languages. The proof of the non-definability of free indexation actually yields a stronger result—consistency of theories that employ free-indexation is, in general, undecidable. As a result, it may be impossible to determine exactly what the consequences of such a theory are. These results call into question the appropriateness of free-indexation as a component of linguistic theory, a question that has been raised on linguistic grounds elsewhere in the literature.

Of course complexity results are not the typical reasons one undertakes a formalization of a theory. The main thing one gets from a formalization is the ability to carry out formal proofs of the consequences of the theory. One may, for instance, be able to formally establish the consistency of the theory, or establish the independence of various aspects of the theory. Stabler, for example, considers a number of proofs of the independence and non-independence of various sets of principles in Stabler, Jr. 1992, and, in motivating his formalization, quotes Chomsky, “An attempt at full-scale formalization of the relevant assumptions might be in order, given the level of complexity and the range of material that must be considered” (Chomsky 1982). Most importantly, the consequences of a linguistic theory constitute the predictions the theory makes about natural language. It is these predictions that distinguish a theory from a simple taxonomy. In formalizing the theory, one provides a means of exploring these consequences, and of testing them, in a formal framework.

The act of formalizing a theory, in itself, often illuminates details of the theory that are otherwise obscure. Frequently, assumptions and gaps in the theory show up that were not at all apparent in less rigorous treatments. Further, the formalization may well suggest modifications and extensions to the theory, such as the elimination of components that can be shown to be consequences of the remainder of the theory. In some cases the issues involved in formalizing the theory may even suggest alternative statements of the theory that can be justified on linguistic grounds. While we don't claim any results along these lines, we will offer, in passing, arguments that the formalization we develop can, indeed, support such research.

The considerations that motivate most of the details of our approach are "theory internal" in the sense that they are concerned with specific language complexity results. Absent a claim that these language complexity results characterize natural language there is no particular reason to believe that the issues we encounter will be significant for GB in general. It's somewhat surprising, then, to find parallels to these issues arising in the current GB research. Indeed, our success in this program is built largely upon the theoretical work of Rizzi and others which, while motivated by linguistic considerations, addresses our needs almost directly. The strength of these connections is enough to suggest that there are deeper parallels between the intuitions driving that research and the language complexity considerations driving ours.

7.1 The Generative Capacity of GB

There are two types of language complexity results one might establish for GB. First, it may be possible to determine the complexity of the sets of phrase markers licensed by the GB mechanism under some specific set of principles and parameters. The result would characterize the class of natural languages to the extent that those principles capture that class. Alternatively, one might be able to identify some class of principles that can be shown to be sufficient to capture natural languages, and then establish that the languages that can be generated by the GB mechanism when restricted to principles in that class fall within some complexity class. The first type of result gives the generative capacity of some specific theory within the GB framework, where that generative capacity is just the class of languages one gets by varying the parameters through their ranges. It says nothing about the complexity of languages generated by extensions to that theory. The second type is a generative capacity result of a more typical nature. It establishes an upper bound on the complexity of the languages defined by the mechanism no matter what principles (in the restricted class) embody the theory.

One approach to the formal characterization of the set of structures generated by the GB mechanism is due to Berwick. In Berwick 1984, he defines, inductively, the set of trees admissible under GB by specifying a local set as a base and defining a specific set of transformations to the trees in that set. He uses this as the foundation of an argument that the GB languages are recursive. The problem, as pointed out by Perrault 1984, is the difficulty of showing that this formalization actually captures the range of GB languages. This seems, rather, to be a result of the first type. It captures some specific set of principles and, in fact, as Perrault notes, does not address many fundamental principles and phenomena, among these case theory, Theta-theory, passives and raising. Presumably, these could be added, but the result would not necessarily hold for the extended theory.

The central result of this half of the book is also a result of the first type. We show that the sets of phrase markers licensed in GB by a specific set of principles are strongly context-free by showing that those principles are definable in $L^2_{K,P}$. This approach has several advantages over more traditional approaches like that of Berwick 1984. Foremost among these is the fact that our proofs deal directly with the principles in a transparent way. Extensions to the theory in the form of new principles or modifications of those principles already accounted for require only that we provide a translation of those principles into $L^2_{K,P}$ that is consistent with the existing definitions. Further, we have non-definability results for $L^2_{K,P}$ that rule out the possibility of capturing certain types of principles in $L^2_{K,P}$. Thus, in addition to establishing definability for a particular set of principles, we also have an indication of what the limits of that definability are.

This suggests the possibility of establishing a result of the second kind. We assume, on the basis of the evidence cited earlier, that natural languages, as a class, are not context-free. We expect, then, that there will be principles which are necessary to their definition in GB that cannot be captured in $L^2_{K,P}$. Nonetheless, if descriptive complexity results similar to ours can be established for larger language complexity classes (for the indexed languages, for instance) it may be possible to make a realistic argument that GB can be restricted to principles that are definable in the corresponding theory without losing the ability to account for the full class of natural languages. In this way we could establish non-trivial generative capacity results for GB as a whole.

7.2 Formalizing GB via Logic

The body of our approach is a logical formalization of GB (or rather, of a specific theory of English syntax within the framework of GB). The most extensive studies of this sort have been in the area of **principle-based parsers** (Berwick et al. 1991), particularly those of Johnson 1989, Fong 1991, and Stabler, Jr. 1992). There are some significant contrasts between these formalizations and ours. In particular, the universe of their definitions consists of individuals and trees implemented as lists. Predicates are defined recursively over these lists. We are confined to $L^2_{K,P}$. Our universe is individuals and arbitrary sets of individuals, and our language is restricted in that all non-monadic predicates other than those of parent, domination, and left-of must be defined explicitly. Thus, there will be significant differences in both the content and form of our definitions of the same principles. Nonetheless, these formalizations have much to offer our efforts both in precise interpretation of principles and in demonstrating successful approaches to the problems of capturing them in logic.

One lesson to be gained is the suggestion that our goal might be attainable for some significant set of principles. Johnson, for instance, notes that X-bar theory, Theta-theory, and the case filter all seem to involve strictly local principles and could be modeled as tree automata (Johnson 1989). Berwick (1991) points out that a similar conclusion can be obtained from Correa's implementation of a principle-based parser as an attribute grammar (Correa 1991), in that his grammar appears to be of a restricted form known to generate only recognizable sets. Thus, for this fragment of the theory at least, we expect to encounter no difficulty.

The approach to a logical formalization of GB that is closest to ours is due to Marcus Kracht (1995). Here Kracht sketches a formalization of portions of Rizzi's *Relativized Minimality* theory in the *orientation language*, a fragment of dynamic logic which he shows to be compilable into a GPSG-style grammar. While his results are similar to ours—Relativized Minimality, to the extent that he captures it, must license a recognizable set—his formalization is developed as an example, rather than as a primary result, and, consequently, is no where near as comprehensive as ours. Some of the points he does not treat he leaves as open questions. The formalization we give here explicitly addresses some of those questions. In particular, we address the question of whether the multiple levels of structure assumed in GB can be collapsed into a single tree without materially affecting the theory, the related issue of the problems created for such interpretations of the theory by the interaction of movements (which are typically addressed via reconstruction), and, perhaps most importantly, the question of what aspects of GB theories raise the languages they license beyond the class of context-free languages.

7.3 Overview

In the chapters that follow we apply definability in $L^2_{K,P}$ to the problem of determining the language complexity of the sets of trees licensed, within the GB framework, by a number of commonly identified principles. The underlying X-bar structure—roughly the base component of

GB grammars—is simple enough that its definability is nearly trivial. Our results, then, will be concerned primarily with the definability of the principles that constrain that structure. We have, then, a measure of the complexity of various sets of principles—if they are definable within $L^2_{K,P}$ they can be enforced by a CFG, if they are provably non-definable, then they are non-context-free.

We begin, in the next chapter, with a simple survey of the basic structures of GB, followed by a brief discussion of approaches to capturing GB logically and a sketch of our overall approach. We then turn to the issue of non-definability. Here our main result is that indexation, as it is usually employed, is not definable in $L^2_{K,P}$. As indexation is used for a wide range of purposes within the various modules of GB, this establishes the basic theme of our study—since principles that are stated in terms of indexation are not directly definable in $L^2_{K,P}$, the main issue for us is to identify which principles necessarily employ indexation and, where it is not necessary, to reformulate the principles without it. This, as it turns out, is an issue that has appeared, at least indirectly, in a number of current refinements of GB theory. Thus, we are not left to explore this unguided.

Some uses of indexation plainly involve simple structural relationships that are easy to capture—subject-verb agreement, for instance. The two uses that are most interesting theoretically are the use of indices in the theories of binding and control to identify the reference of nominals, and the use of indexation in the theory of movement to identify chains. The particular theoretical framework we choose to follow is the *Relativized Minimality* theory of Rizzi 1990. This is concerned primarily with the notion of government—the fundamental class of relationships that determine the relevant domain in binding theory and, at least as the theory is developed by Rizzi, the connection between adjacent co-indexed categories in chains. Significantly for us, these are relationships with a strictly bounded domain and are amenable to definition in $L^2_{K,P}$. Thus, for the most part, the task here is to capture Rizzi's definition of the government relationships and his re-interpretation of the standard theory in terms of these relationships. Since Rizzi was motivated by linguistic issues rather than any particular desire to eliminate indexation, this is not always a straightforward process. Nonetheless, we are able to capture a great deal of this theory within $L^2_{K,P}$.

In Chapter 10 we begin our formalization with the basic structures and relationships of X-bar theory. We then introduce Head-Government, following Rizzi (Section 10.7). In Chapter 11 we discuss our formalization of the lexicon. In our interpretation, this has an extensive role. It is here that issues of subcategorization, Theta theory and case theory are handled.

We begin our exploration of indexation issues, in Chapter 12, with binding and control—the simpler cases. In treating this, we identify a distinction between those aspects of these theories that have consequences for the distribution of nominals (Principle A and Obligatory Control) and those that concern only the proper interpretation of those nominals (Principles B and C and Optional Control). Without access to indexation, or an equivalent mechanism, we cannot in general enforce specific interpretations. Thus, we cannot enforce this latter class of principles, although we never license a tree in which there is no interpretation consistent with these theories.

The largest part of this half of the book covers our formalization of the theory of movement, Chapters 13 and 14. After introducing the basic notions and surveying the classes of movement in English, we follow, for the most part, Rizzi's development of the theory. One interesting aspect of our interpretation of the theory is that, while we end up with a reasonably complex definition of chains, the *Empty Category Principle* reduces to a requirement of proper-head-government and a simple principle requiring every category to be a member of some (possibly trivial) well-formed chain. The formalization we develop in this chapter accommodates essentially all of the simple forms of movement in English.

The theory developed in Chapter 13 is incomplete in that it cannot account for cases where portions of chains are moved by subsequent movement of the phrases in which they occur. This complication is a problem that must be resolved by every declarative interpretation of movement. The usual approach is via *reconstruction*—effectively undoing the movement. In Chapter 14 we develop a mechanism that resolves this issue purely declaratively.

Finally, in Chapter 15 we discuss the nature of two types of structures for which our interpretation fails: particular analyses of cross-serial dependencies in Dutch, and long-distance extractions in Swedish. In both these cases, the failure can be traced to a violation of properties of movement in English on which our interpretation depends. This leads us to suggest a principle which appears to distinguish the context-free non-context-free and GB languages. We close with some summary remarks in Chapter 16.

While this study is not a tutorial, it is intended to be accessible to those with little or no prior knowledge of GB. Indeed, those familiar with GB will likely find some of our exposition tedious, we hope not intolerably. The accounts we give of the phenomena the theory treats serve not only to explain the principles comprising the theory, but also to fix the specific formulations of those principals that we capture.

8

The Fundamental Structures of GB Theories

In this chapter, we survey the basic structures and relationships that provide the framework for GB. For the most part the discussion will be based on the exposition in Haegeman 1991 and Radford 1988, as well as, to a lesser extent, primary sources (Chomsky 1986 and others cited in the text) and some additional secondary sources—in particular the discussions of Stabler, Jr. 1992, Fong 1991, and Johnson 1989.

8.0.1 Levels of Representation and X-Bar Theory

Government and Binding Theory is generally defined in terms of four levels of representation (Figure 3); each sentence is analyzed as four distinct syntactic structures which are related in specific ways. **D-Structure** corresponds roughly to the deep-structure—the output of the base grammar—of Transformational Grammar. All trees generated here share an extraordinarily simple configuration—the basic X-bar structure (Figure 4). Phrases have three levels: the phrase itself, or **maximal projection** (XP or $\overline{\overline{X}}$), consisting of a possible **specifier** (as in the determiner of a NP) and an $\overline{\mathrm{X}}$ head, which in turn consists of some set of **complements** (as in the arguments of a verb) and the head of the phrase (X^0 or just X). These are referred to as bar levels 2 through 0, respectively. Specifiers and complements are required to be maximal projections. In keeping with the *projection principle* selection of specifiers and complements is determined by the lexical head of the phrase, that is, by the content of the head. For predicates, this selection is closely related to argument structure. It is at D-Structure that Theta-roles are assigned. In addition to the lexical categories (N, V, A, and P) GB employs the functional categories INFL (I), inflection—carrying agreement and tense, and COMP (C), a (possibly empty) complementizer. Clauses ($\overline{\mathrm{S}}$ in prior terminology) are analyzed as CPs, projections

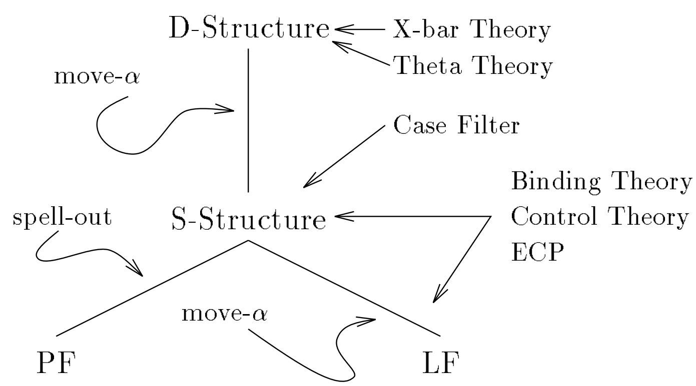

FIGURE 3 Levels of Representation

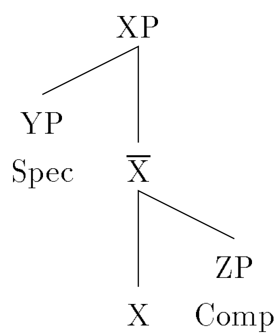

FIGURE 4 X-bar Structure

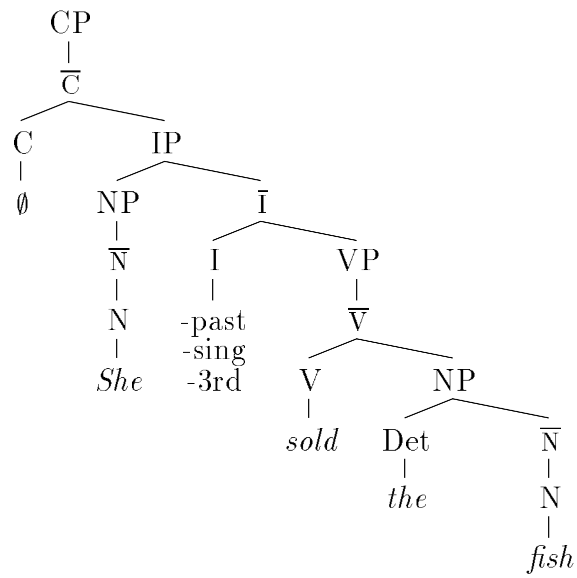

FIGURE 5 Structure of a Typical English Sentence

of C (whether empty or not). Propositions (S) are taken to be IPs. COMP selects an IP complement, INFL usually selects a VP (or AP). Figure 5 demonstrates the X-bar structure of a typical English sentence.

The structure of IP is sometimes analyzed further, following Pollock (1989), as a complex of phrases headed by T^0 (tense), possibly Neg^0 (negation), and Agr^0 (agreement).[1] The same sentence in this interpretation is diagrammed in Figure 6.[2] While there are arguments that turn on this refinement, it is often more convenient to work with the simple INFL. Thus, they are often employed side-by-side, as the need arises, in which case INFL might be regarded as a notational convenience. Our interpretation is neutral to this issue. As we will see when we define the lexicon, either analysis, or even both, can be accommodated depending only on the details of how the lexicon is defined.

The underlying D-Structure is permuted by move-α to produce S-Structure.[3] This is as close as we will get to the surface form of the sentence. As a rule the lexical content at this structure is devoid of

[1] There is some controversy over the relative order of TP and AgrP. Pollock has TP contain AgrP. Rizzi 1990 requires the order in the figure.

[2] Other finer analyses have been proposed as well.

[3] Since move-α is a reversible process a more general way of thinking of this is that D-Structure and S-structure are *related* by move-α.

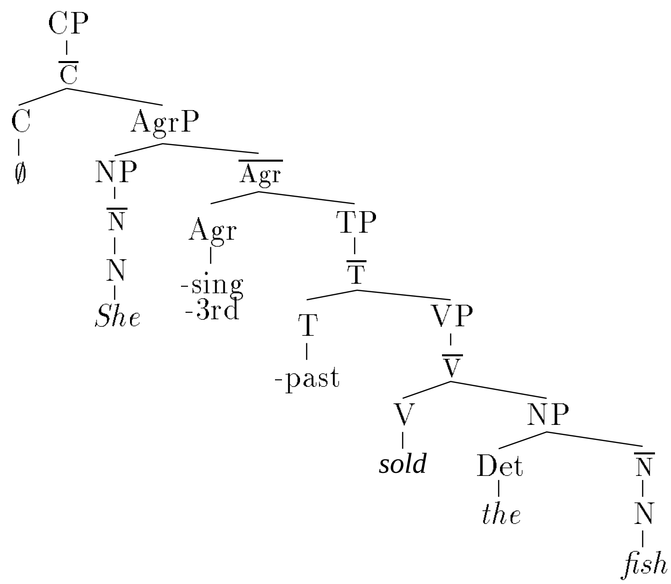

FIGURE 6 An Alternative Structure

inflection, contractions, etc. These are produced by the "spell-out" process that relates S-Structure to PF, Phonetic Form, the actual surface structure. For inflections at least, an alternative analysis is possible, in which the lexical content at S-Structure is fully inflected and the spell-out process is one of checking agreement of that inflection with INFL (see, for instance, Chomsky 1993). While it is largely immaterial to our program, we will adopt this second approach as a matter of convenience. Principles associated with Case Theory, Binding Theory (with some controversy), and constraints on movement, such as subjacency and at least some aspects of the Empty Category Principle (ECP), apply at S-Structure.

Finally, LF, Logical Form, encodes the logical structure of the utterance. This is related to S-Structure by move-α. The associated transformations, for the most part, have to do with raising *wh*-elements and quantifiers to the leading edge of their scope. The traditional approach to constraining movement with ECP requires it to ultimately apply at LF. For most of the analyses we cover, the distinction between S-Structure and LF is not significant; our formalization focuses on D-Structure and S-Structure and does not cover LF.

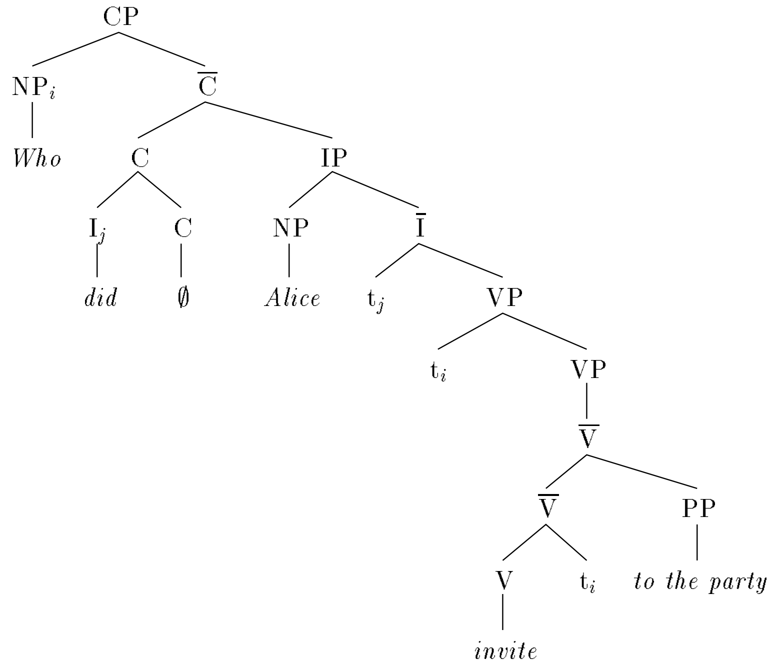

FIGURE 7 Adjunction and Chains

8.0.2 Variations on X-Bar Structure

The **Structure Preserving Principle** requires X-bar structure to be respected at all levels of representation. However, the basic structure is modified, in practice, in two significant ways. First nodes may be **adjoined** (i.e., Chomsky adjunction) either to the right or left of another element (Figure 7). In the figure, I_j is adjoined to the left of C and PP is adjoined to the right of $\overline{\mathrm{V}}$. In this configuration the element at which the adjunction occurs is split into two **segments** (or more if there are multiple adjunctions). Collectively, these segments form a **category**. For the purposes of the theory we do not want to distinguish between the segments of a category. Thus, categories, rather than nodes, are the atomic objects of the theory.

Adjoined categories occur under two circumstances. The first, **base generated** adjunction occurs in D-Structure and accounts for modifiers (adjuncts). Our interpretation of the theory makes no assumption about where base-generated adjunction can occur. It is, however, generally stipulated that only maximal projections can be adjoined in this way. The second source of adjunction is adjunction that occurs as a result of movement (**movement generated** adjunction). This is generally restricted to the cases of heads adjoining to heads (X^0 to Y^0) or maximal projections adjoining to maximal projections (XP to YP).

Movement need not create adjunction structures. Frequently a category will move by **substitution** to a node that was left open at D-Structure. In the figure, the *wh*-NP has moved, first by adjunction to VP and then by substitution at the specifier of CP ([Spec,CP]). Note that the structure preserving principle implies that XPs can only substitute at YP positions and X^0 can only substitute at Y^0 positions.

The second modification to X-bar structure is a consequence of the nature of move-α. When a category is moved it leaves a **trace** behind in its original position. These traces have no phonological content and no constituents, but they do have syntactic features, sharing certain of the features of the moved category and exhibiting some of their own. As movement can occur in several steps (cyclically) each moved category may have several traces. The moved category and its traces together form a **chain** which is usually indicated by co-indexation—marking them with the same index. In the figure the *wh*-NP and its co-indexed traces form a chain, as do the I and its trace. Under some circumstances it is useful to regard members of a chain as a single category, just as segments of an adjunction structure form a single category, but this is by no means standard.

8.1 Representational and Derivational Interpretations

Recent theories frequently do not take the notion of move-α as literal movement. It is often given a **representational** rather than **derivational** interpretation. That is, one can regard move-α and the constraints on movement declaratively, licensing the the existence of chains in S-Structure and LF rather than creating them by a sequence of derivation steps. Under this interpretation the need for multiple levels of representation is no longer obvious. Constraints that apply at D-Structure can be re-interpreted as applying to chains, or applying through the mediation of chains. D-Structure might then be regarded as a perspicuous means of specifying certain constraints on S-Structure and LF.[4] This approach is not without problems. As we will see later, move-α can seriously distort what were compellingly clear conditions at D-Structure. So much so, that these conditions are often assumed to apply under *reconstruction*—as if the checking mechanism applies them in their orig-

[4] Or, carrying this to its logical conclusion, D-structure and S-structure are means of specifying constraints on and the relationship between PF and LF. See Chomsky 1993.

inal form after undoing the movement. (This is an alternate conception of the idea of the condition holding through the mediation of the chain.)

Given this variation in the theoretical conception of movement, it's interesting to see how it plays out in practice in the principle-based parsers. Stabler (1992) takes a derivational approach in the sense that he explicitly defines each of the levels of representation and defines move-α (or its equivalent) as a relation between them. Fong (1991) takes a reconstructive approach. He does not explicitly build D-Structure, but extracts D-Structure relationships (essentially by "de-referencing" chains) when needed. Particularly interesting is Johnson's approach in Johnson 1989. Here he begins with an implementation like that of Stabler, but then applies optimizations in the form of program transformations to this implementation. In doing so first D-Structure and then S-Structure become redundant in the sense that no predicates depend on them. Thus he, in essence, optimizes them out and ends up with, in addition to a more efficient implementation, a representational interpretation.

Our approach is purely declarative. We actually cannot define move-α as an operation on trees or even a relationship between trees. We are forced to adopt a representational interpretation of the principles and to do so at a single level of representation. The representation we define is closest to S-structure. This is mostly a convenience. Being close to PF, the yields of S-structure trees are reasonably familiar. More importantly, LF effects are not critical to the issues we discuss. For reasons we have hinted at, we cannot rely on the mechanism of indexation to identify the members of chains. This prevents us from treating chains, in the manner of Fong, as a sort of indirection for the purposes of checking principles that apply at D-Structure. Consequently, we define our particular variant of S-Structure so that every important D-Structure relationship is preserved directly, by some means, in the S-Structure. Justification for this approach on linguistic grounds can be found in Brody's (1993) argument that D-Structure is best understood as the substructure of S-Structure generated by its restriction to the base positions of chains.

It should be noted that the issue of how transformations are represented is independent of the issue of whether they are an essential component of linguistic theories. This latter issue questions the GB assumption of distinguishable levels of structure—with some linguistic relationships being expressed at one level, others at another, and still others as transformations between the levels. At the other extreme are "non-transformational" theories (such as GPSG Gazdar et al. 1985) in which all linguistic relationships are expressed within a single structure and many of the artifacts of GB transformations, in particular intermediate traces, have no role. In between fall theories like Brody's interpretation of D-Structure or Koster's *Radical Autonomy* (1987), in which the syntactic consequences of the GB theory of movement are accepted but simplifications are found in encoding linguistic relations in a single structure.

The distinction between these two issues becomes clearer once one recognizes that, so far as mathematical structures go, there is nothing extraordinary about transformations. If we take the configuration of Figure 3 literally, then we are concerned with the theory of a class of compound structures including components encoding each of the levels of representation (D-Structure, S-Structure, etc.). The transformations between the levels are just relations that happen to connect individuals in one of these components to individuals in another.[5] As we have seen in the first part of this book, it is not unusual for theories over one class of structures to have *faithful interpretations* into theories over a distinct class—there is a translation between formulae such that a given formula is valid in one class iff its translation is valid in the other. From the point of view of the theory, classes for which such faithful interpretations exist are equivalent. Given the particular characteristics of move-α, it is possible to represent all of the levels of Figure 3 (with the possible exception of PF) within a single tree with each level being the restriction of that tree to a certain subset of its nodes. One can translate faithfully between the theories in terms of these two structures; the distinctions between them have no theoretical weight.

Note that none of the formalizations we have cited actually realizes the structure of Figure 3 directly, even Stabler interprets trees as a particular class of lists. The intent of these formalizations, however, as with ours, is not to modify the theory, but to capture it as faithfully as possible. In contrast, both the non-transformational theories and Koster's reinterpretation of transformations as a class of relations within a single tree offer alternatives to the theoretical formulation of linguistic relationships that is embodied in GB. Thus the arguments for these alternatives, while usually couched in terms of the underlying structures, are not, in fact, about the structures themselves, but are rather about the way in which particular linguistic relationships are represented on those structures.

[5] To see the "ordinariness" of this it perhaps helps to think of rewrite rules as transformations from one level in the derivation tree to the next. If we think, then, of D-Structure as a tree generated by such rules, then we can see it as a compound structure in itself, consisting of a number of levels each related to the next by a specific range of transformations. The full structure, then, just takes this into another dimension.

In closing this discussion, we should note that, primarily because we cannot employ free-indexation, there are distinctions between the way in which we formalize certain relationships and the way in which they are typically defined in GB. In some cases, as in our formalization of the ECP, there is considerable simplification; others are considerably more complex. Given that the role of indexation in linguistic theories is a topic of debate, the ways in which principles must be restated to eliminate it is of interest in its own right, and this study should help to illuminate that issue. On the other hand, we do not presume to propose, in the manner of Koster, any significant modification to the theory—rather, we are reformulating it within the restrictions of $L^2_{K,P}$.

9

GB and Non-definability in $L^2_{K,P}$

Before developing our formalization of those principles that are definable in $L^2_{K,P}$, it will be useful to explore the boundaries of that definability with respect to GB. We have a number of non-definability results from Chapter 4. Some of these have consequences for particular GB analyses.

These results, recall, are based on reductions from non-decidable theories. By the definition of definability, the theory of any class of structures definable in $L^2_{K,P}$ when augmented with any relation that is definable in $L^2_{K,P}$ is still decidable. One way of showing non-definability of a relation, then, is to show that by adding it to $L^2_{K,P}$ one can define a class of structures that has an non-decidable monadic second-order theory. For most of these results, the class of structures involved includes just the grid. This is a structure with two successor functions similar to the infinite binary tree, but in which the right successor of the left successor of a node is equal to the left successor of its right successor, i.e., $x01 \approx x10$. One can define sets in the monadic second-order theory of the grid that correspond to the halting computations of any given Turing Machine. These sets are non-empty, then, iff the given Turing Machine halts. This is the tiling argument of Lemma 1 in Chapter 4. It follows that, if one can capture the theory of the grid in $L^2_{K,P}$ using a predicate for a particular relation, then that relation is not definable in $L^2_{K,P}$. Results based on reduction from the theory of the grid include non-definability of the **equal level predicate**, which is true of a pair iff they are at the same depth in the tree, and non-definability of the predicate $\mathrm{Iso}_{\vec{P}}(x, y)$ which is true of a pair of nodes x and y just in case the subtrees rooted at the nodes are isomorphic with respect to the monadic predicates in $\vec{P}$. We will use this approach in establishing the central non-definability result of this section.

The other reduction we employed in proving non-definability was from the problem of whether the intersection of two CFLs was empty.

The main result here was that the predicate $\mathrm{YieldsEq}_{\vec{P}}(X, Y)$ is not definable in $L^2_{K,P}$. This relation is true of pairs of subtrees iff their yields are labeled identically with respect to $\vec{P}$. In $L^2_{K,P}$ extended with $\mathrm{YieldsEq}_{\vec{P}}$ we can define, given any pair of CFGs, a set of finite trees that is empty iff the intersection of the string languages generated by those CFGs is non-empty. Since emptiness of the intersection of CFGs is non-decidable, emptiness of this set is non-decidable. It follows that $\mathrm{YieldsEq}_{\vec{P}}$ is a not definable in $L^2_{K,P}$.

9.1 Some Non-Definable GB Analyses

As an example of how these results relate to GB, consider the argument of Bresnan, et al., (1982) that a particular analysis of cross-serial dependencies in Dutch is not strongly context-free. They propose a (linguistically motivated) analysis of sentences of the form

...	dat	Jan	Piet	...	de kinderen	zag	helpen	...	zwemmen
...	that	Jan	Piet	...	the children	see-past	help-inf	...	swim-inf

...that Jan saw Piet help ...the children swim

where arbitrarily many noun/verb pairs may be inserted in the ellipses. There are agreement constraints between the nouns and their corresponding verbs. This *is* weakly context-free since all but the first and the last verb/noun pairs share the same agreement conditions, and thus, permuting those nouns or verbs (or both) does not change acceptability of the sentence. The language, then, is closer in nature to $a^n b^n$ than ww; the number of nouns and verbs must match, but, other than the first and last pair, we need not distinguish them.[1] In their proposed analysis, however, the nouns and the verbs occur along separate branches of the tree. Consequently, these branches must be of equal length. The formal part of their argument is very close in its foundation to our approach here. They show that the set of these structures is not strongly context-free using Thatcher's pumping lemma for finite-state tree automata (Thatcher 1967). We can get the same result by noting that the ability to establish that pairs of branches are of equal length gives us the ability to define the equal level predicate, and so ultimately allows definition of the grid. We will return to this example in Chapter 15 when we consider structures that our formalization cannot capture.

Another analysis that we cannot capture directly is **movement by**

[1] This is in contrast to Shieber's Swiss German example Shieber 1985 in which each pair exhibits either of two agreement features resulting in a language similar in nature to $\{ww \mid w \in \{a, b\}^*\}$.

copying. This has been considered in a number of places, most recently in Chomsky 1993. In this account of movement it is proposed that, at the equivalent of S-Structure, a moved category and its trace dominate exactly the same subtree. The ability to enforce this constraint is essentially the ability to enforce the $\text{Iso}_{\vec{P}}$ predicate, and thus, not within the power of $L^2_{K,P}$. Even if one weakens the analysis to require only that the category and its trace dominate the same lexical material, it is still beyond our capacity, since this would entail the $\text{YieldsEq}_{\vec{P}}$ predicate. Note that this says nothing about the non-context-freeness of sets defined by a particular use of this mechanism. It says only that the mechanism is powerful enough to define sets that are not context-free. An argument that some set that has been defined using this mechanism can be captured in $L^2_{K,P}$ is, in essence, an argument that the mechanism is not used in that definition in a necessary way.

9.2 Non-Definability of Free-Indexation

Our examples so far are not very compelling. We don't expect to be able to handle cross-serial dependencies in general, and it is not generally assumed that movement necessarily involves copying. The main result of this section, though, concerns the non-definability of a mechanism that appears in nearly all aspects of GB theory—indexation. We have already mentioned that chains are identified (in part) by co-indexation of their members. Indexation is also used to indicate co-reference in the theories of Binding and Control. That is, an anaphor or pronominal and its referent are marked with a common index. In addition, it is usually assumed that agreement constraints between categories are indicated by co-indexation.[2] In GB, assignment of indices is often assumed to occur by a process of **f**ree-indexation—indices are assigned to categories randomly and those assignments that do not meet the various constraints on chains, binding, control, or agreement are filtered out. In essence, the indexation is an equivalence relation with unbounded index (it distinguishes unboundedly many equivalence classes). That is, each value of the index identifies an equivalence class, and there can be no a priori bound on its maximum value. Free-indexation views constraints on the indexation as a filter that admits only those equivalence relations that meet specific conditions on the individuals in each equivalence class.

To see that such a relation is not definable in $L^2_{K,P}$, suppose we extend $\mathcal{N}_2$ with indices and a predicate $\text{CI}(x, y)$ which holds if and only

[2] While these uses are functionally distinct, there are occasions where an analysis is based on conflating them, although this never seems necessary.

if x and y are co-indexed. We show that this results in a non-decidable theory.

Let

$$\boldsymbol{\mathcal{T}}_{CI} = \{\langle T_2, \triangleleft^*, \leq, \mathrm{r}_0, \mathrm{r}_1, \mathrm{CI}\rangle \mid \mathrm{CI} \text{ is an equivalence relation on } T_2\},$$

that is, the class of structures built by extending $\mathcal{N}_2$ with a single index and a binary relation that holds just in case its arguments are co-indexed. Let S2S+CI be the monadic second-order theory of $\boldsymbol{\mathcal{T}}_{CI}$.

Theorem 1 S2S+*CI is not decidable.*

We will prove this by reduction from the theory of the grid. The proof is nearly identical to the proof of non-definability of $\mathrm{Iso}_{\vec{P}}$ in Section 5.3.5 (Theorem 2 of Chapter 5). We do not repeat here the definitions and lemmas they have in common. The idea is to interpret paths from the root in $\mathcal{N}_2$ as paths (non-decreasing in both x and y) from the origin in $\mathbb{N}^2$ following the same sequence of successors. Of course there are multiple paths to most points in $\mathbb{N}^2$, and these each correspond to a unique point in $\mathcal{N}_2$. We use indexation to identify sets of points in the tree that correspond to the same point in the grid. All that is then required is to insure that the co-indexed points agree on the features relevant to the proof of non-definability of the grid, that is, to insure that, for some set of monadic predicates $\vec{P}$, a point is in some set $P_i \in \vec{P}$ iff all of its co-indexed points are also in that set.

We begin by axiomatizing the appropriate indexation relation. To simplify the connection with $\mathbf{Th}_2(G)$, we use the original language of S2S.

$$
\begin{aligned}
&\Phi_G(\vec{P}) \equiv \\
&\quad (\forall x, y)[\ \mathrm{CI}(x, y) \leftarrow\ \ (x \approx y \vee \\
&\qquad\qquad (\exists x', y')[\ \mathrm{CI}(x', y') \wedge \\
&\qquad\qquad\quad ((x \approx x'0 \wedge y \approx y'0) \vee \\
&\qquad\qquad\quad (x \approx x'1 \wedge y \approx y'1) \vee \\
&\qquad\qquad\quad (x \approx x'10 \wedge y \approx y'01))\,])\quad \wedge
\end{aligned}
\tag{1a}
$$

—x **and** y **are co-indexed if equal,**
if similar children of co-indexed nodes,
or if one is left-child of the right-child
and the other right-child of the left-child
of co-indexed nodes.

$$\mathrm{CI}(x, y) \rightarrow \mathrm{Agree}_{\vec{P}}(x, y, \vec{P})\quad] \tag{1b}$$

—**Co-indexed nodes agree on** $\vec{P}$**.**

Where

$$\mathrm{Agree}_{\vec{P}}(x, y, \vec{P}) \equiv \bigwedge_{\mathrm{P}_i \in \vec{P}} (\mathrm{P}_i(x) \leftrightarrow \mathrm{P}_i(y)). \tag{2}$$

Recall, from Section 5.3.2, that the proof of non-definability of $\mathbf{Th}_2(G)$, the monadic second-order theory of the grid, was based on the fact that

$$(\exists \vec{D})[\phi^{\mathcal{D}}_O \wedge \phi^{\mathcal{D}}_H \wedge \phi^{\mathcal{D}}_V \wedge \phi^{\mathcal{D}}_\tau] \in \mathbf{Th}_2(G)$$

iff there is a tiling accepted by the tiling system $\boldsymbol{\mathcal{D}} = \langle \mathcal{D}, \mathcal{D}_0, \mathcal{H}, \mathcal{V}\rangle$, and that such tiling systems could encode the set of accepting computations of arbitrary Turing Machines. Here $\phi^{\mathcal{D}}_O$, $\phi^{\mathcal{D}}_H$, and $\phi^{\mathcal{D}}_V$ encode $\mathcal{D}_0$, $\mathcal{H}$, and $\mathcal{V}$, respectively. It should be noted that these formulae are built solely from atomic formulae involving O, the successor functions r_0 and r_1, the monadic predicate variables in $\vec{D}$, and the logical connectives.

We claim that the conjunction (roughly) of a formula in this class with $\Phi_G(\vec{D})$ is in S2S+CI iff that formula is in $\mathbf{Th}_2(G)$ (allowing for the translation $\mathrm{O} = \boldsymbol{r}$). In particular, we claim that

$$(\exists \vec{D})[\phi^{\mathcal{D}}_O \wedge \phi^{\mathcal{D}}_H \wedge \phi^{\mathcal{D}}_V \wedge \phi^{\mathcal{D}}_\tau \wedge \Phi_G(\vec{D})]$$

will be in S2S+CI iff

$$(\exists \vec{D})[\phi^{\mathcal{D}}_O \wedge \phi^{\mathcal{D}}_H \wedge \phi^{\mathcal{D}}_V \wedge \phi^{\mathcal{D}}_\tau]$$

it is in $\mathbf{Th}_2(G)$.

Note that, as with $\phi^{\mathcal{D}}_O$, $\phi^{\mathcal{D}}_H$, $\phi^{\mathcal{D}}_V$, and $\phi^{\mathcal{D}}_\tau$, the formulae from which

$\Phi_G(\vec{P})$ is constructed involve only relationships (other than CI) between individuals that are quite close in the tree—either an immediate successor or the successor of a successor. These are certainly well within the range of the kinds of constraints on indexation one might expect to find in natural theories of grammar.

The claim will follow from a few lemmas. First we establish that whenever $\langle T_2, \triangleleft^*, \trianglelefteq, \mathrm{r}_0, \mathrm{r}_1, \mathrm{CI}\rangle$ satisfies $\Phi_G(\vec{P})$ with some assignment, then every pair of nodes in T_2 that correspond to the same point in $\mathbb{N}^2$ under the interpretation sketched above will be co-indexed by CI.

Let $\mathcal{E}_G$ be defined, as in Section 5.3.5, as the set of pairs of points in T_2 that correspond to the same point in $\mathbb{N}^2$.

Lemma 2 *Suppose*

$$M = \langle T_2, \triangleleft^*, \leq, \mathrm{r}_0, \mathrm{r}_1, \mathrm{CI}\rangle \in \boldsymbol{\mathcal{T}}_{CI} \text{ and } M \models \Phi_G(\vec{P})[P_i \mapsto P_i^M]_{P_i \in \vec{P}},$$

then $\mathrm{CI} \supseteq \mathcal{E}_G$.

Proof. We must show that if M models $\Phi_G(\vec{P})$ with some assignment, then all pairs $\langle v, w\rangle \in T_2^2$ for which v is a permutation of w will be in CI. We do this by induction on $|v|$, the length of v, which, of course, is also the length of w.

If $|v| = 0$, then $v = w = \epsilon$, $\langle v, w\rangle \in \mathcal{E}$ and $\langle v, w\rangle \in \mathrm{CI}$, by (1a).

Suppose v is a permutation of w, $|v| \geq 1$ and for every pair $\langle v', w'\rangle$ of length less than $|v|$, if v' is a permutation of w', then $\langle v', w'\rangle \in \mathrm{CI}$.

Suppose v and w have the same final element, that is, $v = v'i$ and $w = w'i$ for $i \in \{0, 1\}$. Then v' is a permutation of w', and by the induction hypothesis $\langle v', w'\rangle \in \mathrm{CI}$. It follows, then, by (1a), that $\langle v, w\rangle \in \mathrm{CI}$.

Suppose, alternatively, that v and w differ on their last element. Without loss of generality, suppose v ends with a '0' and w ends with a '1'. Then, since v is a permutation of w, $v = v'1v''0$ and $w = w'0w''1$ for some $v', v'', w', w'' \in \{0, 1\}^*$. Note that

$$\langle v'1v'', v'v''1\rangle, \langle w'0w'', w'w''0\rangle, \langle v'v'', w'w''\rangle \in \mathcal{E}_G$$

and each of these is of length less than $|v|$. It follows, by the induction hypothesis, that each of these pairs is in CI. Then, by (1a), we have

$$\langle v'1v''0, v'v''10\rangle, \langle w'0w''1, w'w''01\rangle, \langle v'v''10, w'w''01\rangle \in \mathrm{CI}$$

and, by reflexivity and transitivity of CI,

$$\langle v'1v''0, w'0w''1\rangle \in \mathrm{CI}$$

or equivalently, $\langle v, w\rangle \in \mathrm{CI}$. □

The point of this co-indexation is that, via 1b, it requires all nodes in T_2 that correspond to the same node in $\mathbb{N}^2$ to be assigned to the same subsets P_i^M. Thus, just as in the proof of Theorem 2 of Chapter 5 we can take the quotient of M with respect to these classes of nodes without disturbing the assignment. For the class of formulae relevant to the tiling problem this quotient preserves satisfiability, that is, for this class of formulae every $M \in \mathcal{T}_{CI}$ that satisfies $\Phi_G(\vec{P})$ is logically equivalent to M_G.

Lemma 3 *For $\phi(\vec{P})$ in the class relevant to the tiling problem,*

$$M \models (\exists \vec{P})[\phi(\vec{P}) \wedge \Phi_G(\vec{P})] \textit{ iff } M_G \models (\exists \vec{P})[\phi(\vec{P})].$$

The proof is essentially the same as that of Lemma 4 or Section 5.3.5.

It is easy to verify that the model $\langle T_2, \triangleleft^*, \trianglelefteq, \mathrm{r}_0, \mathrm{r}_1, \mathcal{E}_G\rangle \in \mathcal{T}_{CI}$ and that, given any assignment $[P_i \mapsto P_i^G]_{P_i \in \vec{P}}$ mapping $\vec{P}$ to sets of equivalence classes in T_2 wrt $\mathcal{E}_G$, the assignment $[P_i \mapsto P_i^M]_{P_i \in \vec{P}}$ that this induces from $\vec{P}$ to subsets of T_2 is a satisfying assignment for the $\vec{P}$ in $\Phi_G(\vec{P})$ for this model. Thus, we get a corollary.

Corollary 4 *$M_G \models (\exists \vec{P})[\phi(\vec{P})]$ iff there exists some M in $\boldsymbol{\mathcal{T}}_{CI}$ such that*

$$M \models (\exists \vec{P})[\phi(\vec{P}) \wedge \Phi_G(\vec{P})].$$

The completion of the proof is now identical to the final steps of the proof of Theorem 2 of Chapter 5.

Lemma 5 $M_G \cong G$

Corollary 6 $M_G \models (\exists \vec{P})[\phi(\vec{P})] \Leftrightarrow G \models (\exists \vec{P})[\phi(\vec{P})]$

For proofs, see Section 5.3.5.

This suffices to prove the theorem.

Proof. **(Theorem 1)** Given a tiling system $\boldsymbol{\mathcal{D}}$, we have:

$$\begin{array}{lll}
\multicolumn{2}{l}{(\exists \tau)[\tau \text{ accepted by } \boldsymbol{\mathcal{D}}]} & \\
\Leftrightarrow & (\exists \vec{D})[\phi_O^{\mathcal{D}} \wedge \phi_H^{\mathcal{D}} \wedge \phi_V^{\mathcal{D}} \wedge \phi_\tau^{\mathcal{D}}] \in \mathbf{Th}_2(G) & \text{from proof of Lemma 1} \\
\Leftrightarrow & G \models (\exists \vec{D})[\phi_O^{\mathcal{D}} \wedge \phi_H^{\mathcal{D}} \wedge \phi_V^{\mathcal{D}} \wedge \phi_\tau^{\mathcal{D}}] & \text{by def. of } \mathbf{Th}_2 \\
\Leftrightarrow & M_G \models (\exists \vec{D})[\phi_O^{\mathcal{D}} \wedge \phi_H^{\mathcal{D}} \wedge \phi_V^{\mathcal{D}} \wedge \phi_\tau^{\mathcal{D}}] & \text{by Cor. 6} \\
\Leftrightarrow & (\exists M \in \boldsymbol{\mathcal{T}}_{CI})[& \\
 & \quad M \models (\exists \vec{D})[\phi_O^{\mathcal{D}} \wedge \phi_H^{\mathcal{D}} \wedge \phi_V^{\mathcal{D}} \wedge \phi_\tau^{\mathcal{D}} \wedge \Phi_G(\vec{D})] & \text{by Cor. 4} \\
\Leftrightarrow & \neg(\exists \vec{D})[\phi_O^{\mathcal{D}} \wedge \phi_H^{\mathcal{D}} \wedge \phi_V^{\mathcal{D}} \wedge \phi_\tau^{\mathcal{D}} \wedge \Phi_G(\vec{D})] \notin \text{ S2S+CI} & \\
 & & \text{by def. of S2S+CI}
\end{array}$$

Thus, the S2S+CI is non-decidable, by reduction from the halting problem. □

Consequently, arbitrary equivalence relations with unbounded index are not definable in $L^2_{K,P}$ and, equivalently, free-indexation is not definable in $L^2_{K,P}$.[3]

9.3 Discussion

This result is actually considerably stronger than just a proof that free-indexation is not definable in $L^2_{K,P}$. It says that the sets of trees definable in $L^2_{K,P}$ augmented with free-indexation not only include sets that are non-context-free, they include sets for which emptiness is not decidable. Thus, in general, it may not be possible to establish the consistency of linguistic theories expressed in terms of free-indexation, or, even in the case that the consistency of the underlying theory can be established, it may be impossible to determine its consequences. In other words, one can define theories that one simply cannot reason about adequately. Further, the proof is modeled, in some sense, after linguistic usage of indexation. It involves only free-indexation constrained by local conditions—in this case relationships between nodes and one or two levels of predecessor—and agreement between co-indexed nodes. Agreement like this is characteristic of the use of indexation in linguistic theories. Thus, the definition of $\Phi_G(\vec{P})$ is reasonably natural. Note also, that the result needs second-order quantification only to existentially quantify the labels of the nodes. If one assumes labeled trees, then a first-order language suffices. This suggests that the consistency of logical formalizations of GB that employ free-indexation may, in general, be non-decidable.[4] This is an explicit statement of a fairly common intuition that indexation is a very powerful mechanism, perhaps too powerful to play a fundamental role in language.

Since free-indexation allows the definition of non-context-free languages, it is interesting to consider just how powerful, from the point of view of language complexity, it is. Sets of finite trees that are finitely definable in reasonable logical languages are always recursive, since, given such a definition and a finite tree, the number of formulae in the definition, the number of variables that occur in those formulae, and the number of possible assignments to those variables are all finite and can thus be exhaustively checked.[5] The fact that we can define sets for which

[3] The property of being an equivalence relation, that is, of a relation being reflexive, transitive, and symmetric, is trivially definable in $L^2_{K,P}$. This result, then, is one way of establishing that SnS extended with any (arbitrary) binary predicate is undecidable.

[4] It does not imply, however, that it is necessarily non-decidable.

[5] We are indebted to Jason Eisner and Eric Rosen for pointing out an error in an earlier version of this work, in which we, via an ill-considered argument, concluded that non-recursive sets of trees were definable in $L^2_{K,P}$ plus free-indexation.

emptiness is undecidable, on the other hand, implies that the class of languages we can define is not contained in the indexed languages (for which emptiness is decidable). Since many of the classes of languages falling between the CFLs and the CSLs that have been studied are properly contained within the class of indexed languages, this rules out much of the known terrain between CFLs and CSLs. As it turns out, we can, in fact, capture the class of CSLs in a reasonably straightforward way.

To see that every CSL is the yield of a set of trees that is definable in $L^2_{K,P}$ plus $\mathrm{CI}(x,y)$, recall that every CSL is generated by some grammar in which every rule rewrites exactly one non-terminal, i.e., is of the form

$$\alpha A \beta \longrightarrow \alpha \gamma \beta$$

for some non-terminal A and sequences of terminals and non-terminals α, β, and γ. The idea is that we can use CI to capture the equal-level predicate,[6] and then use the equal-level predicate to define a set of trees that encode the derivations in that grammar.

We can capture the equal-level predicate as follows:

$$\begin{array}{l}(\forall x_0, x_1, y_0, y_1)[x_0 \triangleleft x_1 \wedge y_0 \triangleleft y_1 \wedge \mathrm{CI}(x_0, y_0) \rightarrow \mathrm{CI}(x_1, y_1)] \\ (\forall x, y)[x \triangleleft^+ y \rightarrow \neg \mathrm{CI}(x, y)]\end{array} \tag{3}$$

This is just the implicit definition of the equal-level predicate from Section 5.3.4 without the equivalence relation axioms (which are unnecessary since we have restricted ourselves to structures in which CI is such a relation). These axioms, then, restrict CI to be the equal-level predicate.

With this we can capture the derivations of a given context-sensitive grammar (in the canonical form we assume above) with a set of conditions, each of which is definable in $L^2_{K,P}$ in a straightforward way:

- The root of the tree is labeled with the start symbol.
- Every level that includes some non-terminal includes exactly one non-terminal that is marked for rewriting.
- No node labeled with a terminal is marked for rewriting.
- Every node that is is not marked for rewriting but that is on a level that includes some node marked for rewriting has exactly one child and that child is labeled identically to the node with the possible exception that it may be marked for rewriting.
- For every node that is marked for rewriting there is some rule (as given above) such that:
    - The node is labeled A.

[6] This alone is sufficient to establish non-definability of CI in $L^2_{K,P}$, of course, but the definition involves less natural constraints on the indexation than that of the proof of Theorem 1.

  - The immediate predecessors of the node (wrt left-of) on the same level as the node are labeled α.
  - The immediate successors of the node (wrt left-of) on the same level as the node are labeled β.
  - The children of the node are labeled γ.
- No node included in a level in which every node is labeled with a terminal has any children.

It should be clear that every branch of the tree has the same length and that every level of the tree corresponds to a sentential form of a derivation in the grammar. As the root is labeled with the start symbol and the frontier is labeled with a string of terminals, every tree meeting these conditions corresponds to some derivation in the grammar. It is easy to see, furthermore, that every derivation in the grammar corresponds to some tree meeting these conditions. Thus the language defined by the grammar is exactly the yield of the trees defined by these conditions. We have actually captured the language in a slightly stronger way. While the set we define is not strongly equivalent to the grammar—our trees are not the derivation trees defined by the grammar since every level in our trees corresponds to a sentential form (a property not shared by the derivation trees)—we can recover those derivation trees from our trees by collapsing every path in which at most one node is marked for rewriting into a path of length one.

It is important to recognize that these results only hold for our fairly literal interpretation of the notion of free-indexation. It is certainly possible to define indexation relations, even those with unboundedly many equivalence classes, which don't support definition of unverifiable linguistic theories and which are even definable in $L^2_{K,P}$. The significance of the result is that the usual conception of free-indexation, in which arbitrary indexations are filtered by some set of principles, is likely to be problematic even when those principles are limited to expression of local conditions. If linguistic theories that employ indexation are to be verifiable, they are likely to need stronger restrictions on that indexation than these.

Berwick suggests one such restriction in Berwick 1984. Here, in the discussion of his argument that GB can define only recursive sets, he attributes the fact that it must consequently be weaker than Lexical-Functional Grammar (LFG) to the fact that LFG, via unification, effectively has the power to check similarity of structure over all levels of unboundedly deep hierarchical structures (something like $\mathrm{Iso}_{\vec{P}}$). In GB such similarity need only be checked at the top level of the structures. In the treatment of the Bresnan, et al., analysis of Dutch, he argues, LFG would enforce agreement of the two subtrees by unification of functional structures capturing the agreement constraints along those trees—structures of unbounded depth. A GB account, in contrast, needs only check agreement between co-indexed elements of linear phrase markers. He goes on to speculate that a bound such as this on the complexity of the structures that must be checked may be characteristic of structural constraints on natural languages.

The least natural aspect of the definition of $\Phi_G(\vec{P})$ is the fact that, although the constraints on the co-indexation are local and all co-indexed nodes occur at the same level of the tree, it requires every node in the tree to be indexed. While this is a typical assumption for free-indexation, linguistic theories are generally only concerned with the indexation of particular subsets of the tree. Perhaps if the indexation were restricted to a single level in some way (perhaps to sets of nodes that are pairwise related by left-of—a single horizontal "cut" through the tree) we would no longer be able to define theories for which emptiness was non-decidable using it. It would certainly rule out our definition of $\Phi_G(\vec{P})$.

But even this appears to be too strong. Suppose we restrict the indexing to sets that are pairwise related by left-of, or even stronger, to the yield of the tree. Even with this notion of free-indexation we can define $\mathrm{YieldsEq}_{\vec{P}}(X,Y)$:

$$
\begin{aligned}
&\mathrm{YieldsEq}_{\vec{P}}(X,Y) \equiv \qquad\qquad (4)\\
&\quad (\forall x')[\mathrm{Frontier}(x',X) \rightarrow (\exists! y')[\mathrm{Frontier}(y',Y) \wedge \mathrm{CI}(x',y')]] \quad \wedge\\
&\qquad \text{—}\textbf{Every frontier node of } X \textbf{ is co-indexed with}\\
&\qquad\quad \textbf{exactly one frontier node of } Y\\
&\quad (\forall y')[\mathrm{Frontier}(y',Y) \rightarrow (\exists! x')[\mathrm{Frontier}(x',X) \wedge \mathrm{CI}(y',x')]] \quad \wedge\\
&\qquad \text{—}\textbf{Every frontier node of } Y \textbf{ is co-indexed with}\\
&\qquad\quad \textbf{exactly one frontier node of } X\\
&\quad (\forall x',x'',y',y'')[(\mathrm{Frontier}(x',X) \wedge \mathrm{Frontier}(x'',X) \wedge\\
&\qquad\qquad\qquad \mathrm{Frontier}(y',Y) \wedge \mathrm{Frontier}(y'',Y) \wedge\\
&\qquad\qquad\qquad \mathrm{CI}(x',y') \wedge \mathrm{CI}(x'',y'')) \quad \rightarrow\\
&\qquad\qquad\quad (x' \prec x'' \leftrightarrow y' \prec y'')] \quad \wedge\\
&\qquad \text{—}\textbf{The indexing respects left-of}\\
&\quad (\forall x',y')[(\mathrm{Frontier}(x',X) \wedge \mathrm{Frontier}(y',Y) \wedge \mathrm{CI}(x',y')) \rightarrow\\
&\qquad \mathrm{Agree}_{\vec{P}}(\vec{P},x',y')]\\
&\qquad \text{—}\textbf{Co-indexed nodes agree on } \vec{P}
\end{aligned}
$$

where Frontier(x, X) holds just if x is maximal in X wrt domination

$$\text{Frontier}(x, X) \equiv X(x) \wedge (\forall y)[x \triangleleft^{+} y \rightarrow \neg X(y)].$$

The definition is reasonably transparent. The first two clauses require the indexation to reflect a bijection between the yields of the subtrees X and Y. The third clause requires it to preserve left-of. (This is easy to establish by induction on the number of left siblings.) Finally, the last clause requires co-indexed nodes to be labeled identically. Again, second-order quantification is needed here only to existentially quantify the labels. Thus, the result holds for the first-order theory of labeled trees extended with co-indexing. Recall that the non-definability or YieldsEq$_{\vec{P}}$ was established by reduction from emptiness of intersection of CFLs (and thus, indirectly from PCP). Thus again, sets of trees defined using co-indexation, even in this restricted form, are not just non-context-free, their emptiness is not decidable. So some stronger restrictions on the indexation are apparently necessary.

The approach we take in our formalization, in which we are looking for context-freeness as well as decidability of emptiness, in essence, is to bound the range of the indexing, that is, bound the number of equivalence classes it identifies. This approach is not as restrictive as it might seem. We will discuss this in more detail in the following sections, but bounded indexation suffices for us because we do not require co-referring categories to be co-indexed (rather we only require that trees are ruled out whenever there is no indexing that is consistent with the theory), and because we adopt an analysis in which there are boundedly many chains that can occur in a given local context. This latter condition is shared by Berwick's formalization in Berwick 1984, where he assumes that there can be only a fixed number of landing sites in a given cycle.

Given the non-definability of free-indexation as it is usually used in GB, our investigations become, in part, an exploration of which principles actually require unbounded indexation. These explorations may have significance beyond our narrow focus on context-free principles. As we noted earlier, the appropriateness of indexation as a basic linguistic mechanism has been questioned. Chomsky, in endorsing an account of binding theory that does not make use of indexation, notes

> A theoretical apparatus that takes indices seriously as entities, allowing them to figure in operations (percolation, matching, etc.), is questionable on more general grounds. Indices are basically the expression of a relationship, not entities in their own right. They should be replaceable without loss by a structural account of the relation they annotate. Chomsky 1993, pg.49, note 52

We are compelled to look for these structural relationships underlying uses of indexation in linguistic theory by the fact that they are the only types of relationships we can capture. Chomsky and others questioning these uses of indexation have been led to explore the same issues by more purely linguistic considerations. This is one of the cases in which our research, motivated by language complexity considerations, seems to converge with research motivated by such linguistic considerations.

10

Formalizing X-Bar Theory

We begin our formalization with X-bar structure. The sets of trees we define are essentially S-Structure trees, and our interpretation of X-bar reflects this. We take the representational approach and regard move-α and conditions on chains as constraints on S-Structure. While we will not concern ourselves yet with the implementation of those constraints, the structure we define allows for traces and both base-generated and movement-generated adjunction. We allow adjunction of X^0s to other Bar0 nodes. In fact, for reasons we will explain shortly, we assume that all head movement is adjunction. We start with definitions of the basic constituents and structural relations of the X-bar scheme.

We will be defining four classes of predicate: properties of nodes in the tree (or equivalently, the assignment of simple features to nodes), relations between nodes, properties of subsets of the tree, and relations between those subsets. As we discussed in Chapter 4, only properties of nodes have actual interpretations in $L^2_{K,P}$—as existentially quantified second-order variables. We can think of these predicates as labeling the tree. Our definitions license particular distributions of these labels. The other classes of predicate must be defined using *explicit definitions* (which we denote with $\equiv$), that is, definitions that ultimately resolve into formulae involving only monadic first-order predicates and the dyadic predicates of the signature ($\triangleleft$, $\triangleleft^*$, and $\prec$). These are best thought of as notational conveniences which are expanded by simple substitution of the right-hand side for the left. Thus, care must be taken to avoid circularities in their definition.

10.1 Categories

As we noted in Chapter 8, GB structures are defined in terms of *categories* rather than nodes. That is, the structures do not distinguish between nodes that have been split by the process of adjoining another node to them. We define categories in two steps. A **component** is a sequence of nodes that are segments of the same category, that is, they are connected by parent and linearly ordered by domination, they all share the same set of features, and each node except the minimum segment (the maximum wrt $\triangleleft^*$) is binary branching with exactly one child that is an adjoined node and another that is a member of the component. At this point we make no restriction on the adjoined node. Various constraints on adjunction will be added later. A category, then, is just a maximal component—one that cannot be extended in any way.

$$
\begin{array}{l}
\text{Component}(X) \equiv \\
\quad \text{Path}(X) \wedge \\
\quad (\forall x, y)[X(x) \wedge X(y) \rightarrow \text{F.Eq}(x, y)] \quad \wedge \\
\quad (\forall x, x')(\exists y)(\forall z) \ \ [\quad X(x) \wedge X(x') \wedge x \triangleleft x' \rightarrow \\
\qquad\qquad\qquad\qquad\qquad \neg\text{Adj}(x') \wedge x \triangleleft y \wedge y \not\approx x' \wedge \text{Adj}(y) \wedge \\
\qquad\qquad\qquad\qquad (x \triangleleft z \rightarrow z \approx x' \vee z \approx y) \qquad\qquad]
\end{array}
\tag{1}
$$

$$
\begin{array}{l}
\text{Category}(X) \equiv \\
\quad \text{Component}(X) \wedge \\
\qquad (\forall Y)[\text{Subset}(X, Y) \wedge \neg\text{Subset}(Y, X) \rightarrow \neg\text{Component}(Y)]
\end{array}
\tag{2}
$$

The predicate Subset is defined in Chapter 4. The predicate $\text{F.Eq}(x, y)$ enforces agreement between the segments of the category on all linguistic features. This includes features such as (linguistic) category and whether the category is in its base generated position, etc., but does not include all features. In particular it does not include Adj (discussed momentarily). Exactly which features are shared by the segments is not important here. $\text{F.Eq}(x, y)$, then, might be defined, in part:

$$
\text{F.Eq}(x, y) \equiv (\text{N}(x) \leftrightarrow \text{N}(y)) \wedge (\text{V}(x) \leftrightarrow \text{V}(y)) \wedge \cdots \tag{3}
$$

We use the feature Adj to distinguish adjoined nodes. Its distribution is determined by definitions 1 and 4. This is the first of a great many artificial features that are required to make the formalization go through. We attach no linguistic significance to them; they are simply bookkeeping measures, not altogether different than the slashed categories of traditional CFG treatments of movement. It is possible, if one likes, to distinguish features with linguistic significance from those purely internal to the formalization. We leave the issue of which are which open.

The definition of component requires that every adjoined node is marked Adj and that every sibling of an adjoined node is not. Note that

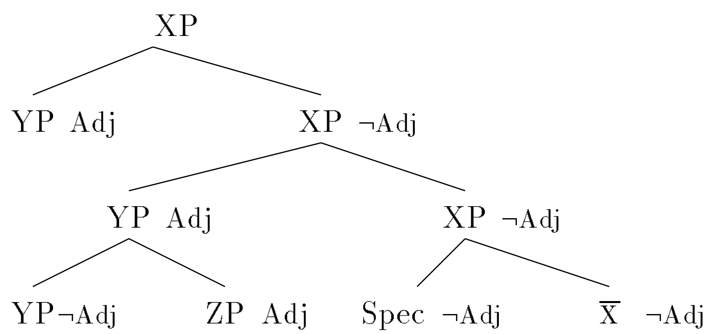

FIGURE 8 An Adjunction Structure

the maximal segment of a category can be either Adj or not depending on whether the category is adjoined to another. We must require, as well, that no node is marked Adj unless it is the child of a non-minimal segment of a non-trivial category:

$$(\forall x)[\quad \mathrm{Adj}(x) \rightarrow (\exists y, z, Y)[y \triangleleft x \wedge y \triangleleft z \wedge \mathrm{Category}(Y) \wedge Y(y) \wedge Y(z)]] \tag{4}$$

An example of an adjunction structure is given in Figure 8. The three XPs are a category that has been formed by adjunction of two YPs. Of these, the lower YP is also a non-trivial category, formed by (right) adjunction of a ZP. The minimal node of this YP category is labeled Adj, since it is adjoined to the XP. The minimal node of the XP category may be labeled Adj or not, depending on whether it is, in turn, adjoined to another category.

Note that every node is a member of some (possibly trivial) category. Further, every component can be extended towards the root in at most one way (with its parent) and, because of the Adj feature, every component can be extended towards the frontier in at most one way as well. It follows, then, that every node is a member of a unique category. We overload the predicate Category to pick out the category of a given node, and those nodes that share the same category.

$$\mathrm{Category}(X, x) \equiv \mathrm{Category}(X) \wedge X(x) \tag{5}$$

$$\mathrm{Category}(x, y) \equiv (\exists X)[\mathrm{Category}(X, x) \wedge \mathrm{Category}(X, y)] \tag{6}$$

To pick out the maximal and minimal segments (nodes) of a category:

$$\mathrm{MaxSeg}(x) \equiv (\exists X)[\mathrm{Category}(X, x) \wedge (\forall y)[X(y) \rightarrow x \triangleleft^* y]] \tag{7}$$

$$\mathrm{MinSeg}(x) \equiv (\exists X)[\mathrm{Category}(X, x) \wedge (\forall y)[X(y) \rightarrow y \triangleleft^* x]] \tag{8}$$

10.2 Basic Structural Relationships

The pattern, as in the definitions of Category, of having multiple predicates that differ only in the types of their arguments is typical of our approach. While categories are the atomic objects of X-Bar theory, the variables of our language range over arbitrary subsets rather than categories. Thus, definitions of predicates in terms of categories tend to be cluttered by qualifications restricting their domain to those subsets that are categories. Rather than propagate these throughout the formalization, we localize them, for the most part, in this section and the next. Relations between categories will generally be expressed as relations between the segments (nodes) of those categories, that is, rather than defining a predicate $\mathrm{R}(X, Y)$ between categories (or, perhaps, in addition to defining such a predicate) we will define a predicate $\mathrm{R}(x, y)$ which holds for every pair of nodes x, y that are segments of categories X and Y, respectively, for which $\mathrm{R}(X, Y)$ is true. As we invariably use upper case for set variables and lower case for individual variables, there should be no confusion between the types of these predicates.

Since categories are the atomic objects of the theory, we need analogs of $\triangleleft^+$, $\triangleleft$, and $\prec$ as relations on categories. As promised, we define them as relations on nodes that hold between the segments of appropriately related categories. A category **dominates** (irreflexive) another just in case every segment of that category dominates every segment of the other. A category **excludes** another just in case none of its segments dominate any segment of the other. It is **left-of** another if the two categories mutually exclude each other and the segments of the first are left-of the segments of the other. Since they exclude each other, no segment of either dominates any segment of the other. It follows that every segment of the one is left-of every segment of the other iff any segment is.

$$\text{Dominates}(x, y) \equiv (\forall x')[\text{Category}(x, x') \rightarrow x' \triangleleft^+ y] \tag{9}$$

$$\text{Excludes}(x, y) \equiv (\forall x')[\text{Category}(x, x') \rightarrow \neg x' \triangleleft^* y] \tag{10}$$

$$\text{Includes}(x, y) \equiv \neg\text{Excludes}(x, y) \tag{11}$$

$$\text{Left-Of}(x, y) \equiv \text{Excludes}(x, y) \wedge \text{Excludes}(y, x) \wedge x \prec y \tag{12}$$

A category **immediately dominates** another if it dominates it and there is no category that falls properly between the two. Immediate domination is used to pick out the children of a category. This is a case where the ambiguous status of adjoined nodes with respect to domination is significant. A category does not dominate, *a fortiori* does not immediately dominate, those categories that are adjoined to it. On the

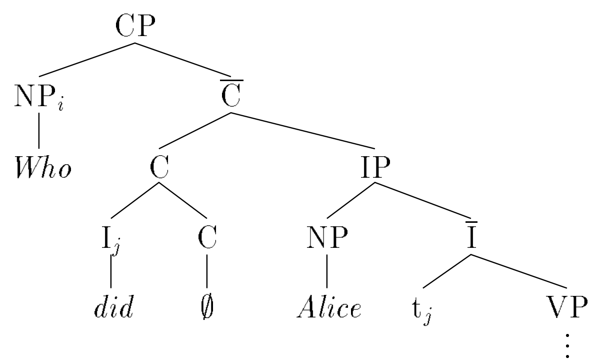

FIGURE 9 Relations between Categories

other hand, we do not want a category to immediately dominate those categories adjoined to its children either; for the purposes of immediate domination the child category must fall between its parent and its adjoined categories with respect to domination.

$$
\begin{aligned}
&\text{Imm-Dominates}(x, y) \equiv \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (13)\\
&\quad \text{Dominates}(x, y) \wedge \\
&\quad (\forall z)[(\text{Excludes}(z, x) \wedge \text{Includes}(z, y)) \rightarrow \text{Category}(z, y)] \\
&\qquad \text{—}\mathbf{no}\ z\ \mathbf{falls\ properly\ between}\ x\ \mathbf{and}\ y
\end{aligned}
$$

Note that the structures defined in terms of categories are richer than simple trees. In a tree, every pair of nodes is related either by equality, proper domination, or left-of. Categories, in contrast, can be related by inclusion without being related by domination. In Figure 9, for example, the category C *includes* the subtree rooted at the highest C. It *excludes* everything else. It *dominates*, on the other hand, only the ∅. I_j is neither dominated nor left-of the category C, nor does it dominate it. This relationship of inclusion without domination plays an important role in GB. It allows I_j to be attached to C without actually being *under* it.

In GB the most important fundamental structural relations are **c-command** and **m-command**. These are relations that correspond roughly to a category being “higher in the tree” than another. There are a variety of variations on the definition of these, or at least on the definition of c-command. We will follow Rizzi (1990) (who in turn is following Sportiche) and say that a category c-commands another iff neither category dominates the other and every category (not just every branching category) that dominates the first dominates the second.

M-command is similar, but it extends to the maximal projection—every XP that dominates the first category must dominate the second. In Figure 9 t_i c-commands VP and m-commands (but does not c-command) NP (*Alice*).

Note that it is possible for a pair of nodes to mutually c-command or m-command each other, as in the case of t_i and VP in the figure. Most commonly, when these relations are employed, the relationship between the relevant categories is actually asymmetric. There are technical reasons for preferring asymmetry,[1] not the least of which is that while asymmetric c-command is transitive, ordinary c-command is not. As an example $\bar{\mathrm{I}}$ c-commands NP in the figure, and NP c-commands VP, but $\bar{\mathrm{I}}$ dominates VP and does not c-command it. We will require asymmetric c-command in our analysis of chains.

$$\begin{aligned}
&\text{C-Commands}(x, y) \equiv \\
&\quad \neg\text{Dominates}(x, y) \wedge \neg\text{Dominates}(y, x) \wedge \\
&\qquad \textbf{—neither dominates the other} \\
&\quad (\forall z)[\text{Dominates}(z, x) \rightarrow \text{Dominates}(z, y)] \\
&\qquad \textbf{—every category dominating } x \\
&\qquad\quad \textbf{dominates } y
\end{aligned} \tag{14}$$

$$\begin{aligned}
&\text{M-Commands}(x, y) \equiv \\
&\quad \neg\text{Dominates}(x, y) \wedge \neg\text{Dominates}(y, x) \wedge \\
&\qquad \textbf{—neither dominates the other} \\
&\quad (\forall z)[(\text{Bar2}(z) \wedge \text{Dominates}(z, x)) \rightarrow \text{Dominates}(z, y)] \\
&\qquad \textbf{—every maximal projection dominating } x \\
&\qquad\quad \textbf{dominates } y
\end{aligned} \tag{15}$$

$$\begin{aligned}
&\text{A-C-Commands}(x, y) \equiv \\
&\quad \text{C-Commands}(x, y) \wedge \neg\text{C-Commands}(y, x)
\end{aligned} \tag{16}$$

The command relations are a case in which the distinction between structures of categories and structures of nodes is significant. These are often defined as relations on nodes in terms of $\triangleleft^+$ rather than Dominates. But the fact that a category is not dominated by the category it is adjoined to is essential to several analyses in GB. The category I_j in Figure 9,

[1] There are linguistic reasons as well. Kayne (1994) employs asymmetric c-command crucially in developing an argument that X-Bar structure can be derived from the assumption that phrase structure uniquely determines the linear order of the terminal string.

for instance, is required to c-command its trace (t_j). This relation holds because c-command is defined in terms of Dominates, and C does not dominate I_j, but this would not be the case if c-command were defined in terms of $\triangleleft^+$.

10.3 X-Bar Structure

Our intention is to capture S-structure. To simplify, we take the basic definition of X-Bar structure absolutely literally. In the literature, non-branching categories are frequently left out of the structure and empty non-lexical heads may be missing. In our interpretation every level is always present including a level for lexical insertion. Our notion of the lexicon includes empty categories such as PRO, *pro*, O, as well as a null element for every head that may be truly empty at S-structure (e.g., COMP). These are necessary because we take the projection principle literally. Each head determines its complements and specifiers;[2] every node is licensed by some head. Thus, in our definition of the lexicon even null heads will be associated with a category feature and formulae selecting complements and specifiers.

It is generally assumed, in accounts of movement in GB, that only XP and X^0 elements may move. We handle movement, in part, using the features Base and Trace. The D-structure position of an element has the feature Base. (This is irrelevant for X'.) Traces have the feature Trace. Treating lack of movement as trivial movement, the target position of any (possibly trivial) chain is that position that is $\neg$Trace. We treat all Head movement as adjunction. The primary reason for this is that substitution of one head for another violates our strict interpretation of X-bar theory at S-Structure. Structurally, adjunction at a null head is nearly identical to substitution. The extreme case for English, V to I to C movement, is shown in Figure 10. Here, again, neither the I nor the C dominate the V. Thus, every node in the complex at the head of CP bears the same c-command and m-command relationships to every node that is not in that complex. We don't claim any linguistic justification or significance for this interpretation of head movement. It is a convenient mechanism for capturing the operations of amalgamation (of I and V) and head substitution (of [V,I] for C) without losing strict X-bar structure.

The $\overline{\mathrm{X}}$ head of an XP y is picked out by $\mathrm{HeadXP}(x, y)$. The head of an $\overline{\mathrm{X}}$ y is picked out by $\mathrm{Head}\overline{\mathrm{X}}(x, y)$. Definition of these, of course, is a

[2] This is an oversimplified interpretation of the projection principle. The actual principles involved in selecting specifiers, in particular, are certainly more complicated than this. These can be captured as refinements to our treatment.

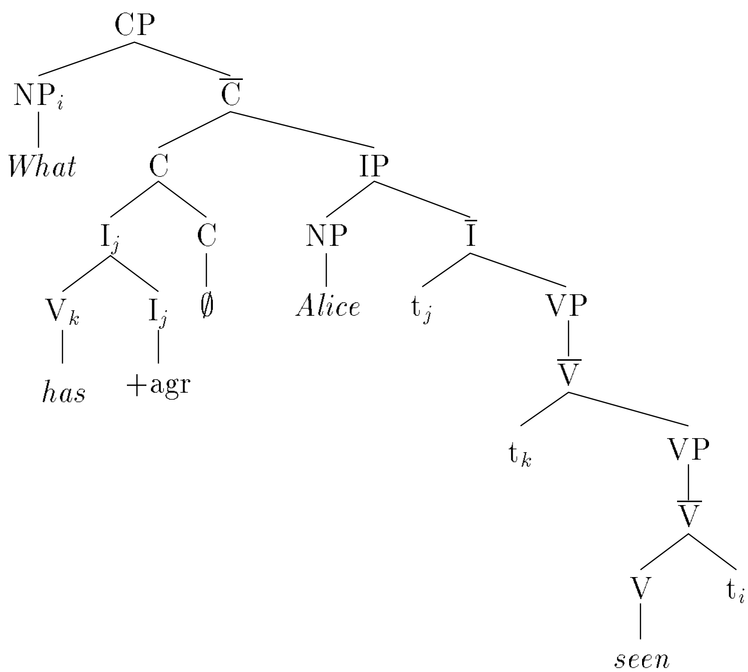

FIGURE 10 V-to-I-to-C Movement in English

parameter of variation. Separating these definitions from the definition of the X-Bar structure itself is one of the few places we follow the notion that GB involves a fixed set of principles specialized with a small set of parameters. In English, specifiers are head-final and complements are head-initial, and so we have:

$$\begin{aligned} \mathrm{HeadXP}(x,y) \quad &\equiv \quad \text{Imm-Dominates}(y,x) \wedge \\ &\quad (\forall z)[\text{Imm-Dominates}(y,z) \rightarrow \neg\text{Left-Of}(x,z)] \\ &\qquad \text{—}x \textbf{ is the right-most child of } y \end{aligned} \tag{17}$$

$$\begin{aligned} \mathrm{Head}\overline{\mathrm{X}}(x,y) \quad &\equiv \quad \text{Imm-Dominates}(y,x) \wedge \\ &\quad (\forall z)[\text{Imm-Dominates}(y,z) \rightarrow \neg\text{Left-Of}(z,x)] \\ &\qquad \text{—}x \textbf{ is the left-most child of } y \end{aligned} \tag{18}$$

These are only valid when x is an XP or $\overline{\mathrm{X}}$, respectively.

Projection of features from a category y to its next higher projection is enforced by the predicate Projects(x,y), which looks something like:

$$\text{Projects}(x,y) \equiv (\mathrm{N}(x) \leftrightarrow \mathrm{N}(y)) \wedge (\mathrm{V}(x) \leftrightarrow \mathrm{V}(y)) \ldots \tag{19}$$

for all relevant features.

All nodes are required either to be at some Bar level, or to be defined in the lexicon.[3]

$$(\forall x)[\mathrm{Bar2}(x) \vee \mathrm{Bar1}(x) \vee \mathrm{Bar0}(x) \vee \mathrm{Lexicon}(x)] \tag{20}$$

XPs may either immediately dominate some (optional) specifiers and an $\overline{\mathrm{X}}$ head or may be traces, in which case they dominate nothing. Since we assume $\overline{\mathrm{X}}$s do not move, these must immediately dominate some (optional) complements and an X^0 head. We must treat X^0 traces differently than XP traces because our strict interpretation of X-bar theory requires an image of the lexical item in its base position in order to select specifiers and complements. Thus, both the target and base positions of head movement immediately dominate a lexicon node. Lexicon items and heads fulfill their X-Bar roles in their base position and not in their moved position. In spell-out the nodes dominated by X^0 traces will be ignored. Again, this is not a linguistically motivated treatment, it is simply a convenient way of simplifying the definition of X-bar structures that have been permuted by movement. Its main consequence here is that all X^0s, even traces, immediately dominate exactly one node and that node is defined in the lexicon. (The requirement that it immediately dominate exactly one node is enforced by $(\exists! y)[\text{Imm-Dominates}(x,y)]$ which implies that y forms a trivial category.) The predicate Base(x) is

[3] This will be modified when we extend the theory to deal with reconstruction in Chapter 14.

true just in the case that x is in its base position. It is defined for the bar level nodes in Chapter 13. The lexical item dominated by the X^0 is in base position iff the X^0 is in base position.

$$
\begin{aligned}
&(\forall x)[\mathrm{Bar2}(x) \rightarrow \\
&\quad \mathrm{Trace}(x) \wedge (\forall y)[\neg \mathrm{Dominates}(x,y)] \quad \vee \\
&\qquad \text{—}x \textbf{ is a trace dominating nothing} \\
&\quad (\exists y)[\mathrm{HeadXP}(y,x) \wedge \mathrm{Bar1}(y) \wedge \mathrm{Projects}(x,y)] \wedge \\
&\qquad \textbf{—or it is the projection of an } \overline{\mathrm{X}} \\
&\quad (\forall y)[(\text{Imm-Dominates}(x,y) \wedge \neg \mathrm{HeadXP}(y,x)) \rightarrow \mathrm{Bar2}(y)] \quad] \\
&\qquad \textbf{—and all non-head children are YPs}
\end{aligned} \tag{21}
$$

$$
\begin{aligned}
&(\forall x)[\mathrm{Bar1}(x) \rightarrow \\
&\quad (\exists y)[\mathrm{Head}\overline{\mathrm{X}}(y,x) \wedge \mathrm{Bar0}(y) \wedge \mathrm{Projects}(x,y)] \wedge \\
&\qquad \text{—}x \textbf{ is the projection of an } \mathrm{X}^0 \\
&\quad (\forall y)[(\text{Imm-Dominates}(x,y) \wedge \neg \mathrm{Head}\overline{\mathrm{X}}(y,x)) \rightarrow \mathrm{Bar2}(y)] \quad] \\
&\qquad \textbf{—and all non-head children are YPs}
\end{aligned} \tag{22}
$$

$$
\begin{aligned}
&(\forall x)[\mathrm{Bar0}(x) \rightarrow \\
&\quad (\exists! y)[\text{Imm-Dominates}(x,y)] \wedge \\
&\qquad \text{—}x \textbf{ has exactly one child} \\
&\quad (\forall y)[\text{Imm-Dominates}(x,y) \rightarrow (\mathrm{Lexicon}(y) \wedge \mathrm{Projects}(x,y) \wedge \\
&\qquad \textbf{—a lexical item that } x \textbf{ projects} \\
&\qquad\qquad\qquad\qquad \mathrm{Base}(x) \leftrightarrow \mathrm{Base}(y))] \quad] \\
&\qquad \textbf{—and is in Base position iff } x \textbf{ is}
\end{aligned} \tag{23}
$$

We define a few more predicates which given any node pick out the components of the phrase immediately including that node. These are false for all nodes if the given node is not in base position unless it is at bar level 2. Note that these predicates are defined for all nodes, those at the lexical level as well as those at bar levels.

Each of these has a monadic version which is true of a node just in case it is the corresponding component of some phrase.

Max-Projection(x,y) holds iff x is a segment of the minimal XP including y. The monadic version Max-Projection(x) is true iff x is an

XP.

$$\begin{aligned}&\text{Max-Projection}(x, y) \equiv \\ &\quad (\text{Base}(y) \vee \text{Bar2}(y)) \wedge \\ &\quad \text{Bar2}(x) \wedge \text{Includes}(x, y) \wedge \\ &\quad (\forall z)[(\text{Bar2}(z) \wedge \text{Includes}(z, y)) \rightarrow \text{Includes}(z, x)]\end{aligned} \tag{24}$$

$$\text{Max-Projection}(x) \equiv \text{Bar2}(x) \tag{25}$$

Head(x, y) holds iff x is the Bar0 level head of the XP containing y. The monadic version Head(x) is true if x is the head of any phrase. Generally, this would be true iff x is Bar0, but we exclude X^0s that are not in base position. We define this explicitly, then, by existentially quantifying the y in Head(x, y). If x is not in base position it cannot be the head of any maximal projection, and there will be no such y.

$$\begin{aligned}\text{Head}(x, y) \quad \equiv \quad & (\text{Base}(y) \vee \text{Bar2}(y)) \wedge \\ & (\exists z_2, z_1)[\text{MaxProjection}(z_2, y) \wedge \\ & \text{Head-XP}(z_1, z_2) \wedge \text{Head-}\overline{\text{X}}(x, z_1)] \\ & \qquad \text{—}z_2 \textbf{ and } z_1 \textbf{ are the XP and} \\ & \qquad\quad \overline{\text{X}} \textbf{ projections of x}\end{aligned} \tag{26}$$

$$\text{Head}(x) \equiv (\exists y)[\text{Head}(x, y)] \tag{27}$$

Comp(x, y) holds iff x is a segment of a complement of the maximal projection of y.

$$\begin{aligned}&\text{Comp}(x, y) \equiv \\ &\quad (\text{Base}(y) \vee \text{Bar2}(y)) \wedge \\ &\quad (\exists z_2, z_1)[\text{MaxProjection}(z_2, y) \wedge \text{HeadXP}(z_1, z_2) \wedge \\ &\qquad\qquad \text{Imm-Dominates}(z_1, x) \wedge \neg\text{Head-}\overline{\text{X}}(x, z_1)] \\ &\qquad\quad \text{—}x \textbf{ is the non-head child of the } \overline{\text{X}} \textbf{ containing } y\end{aligned} \tag{28}$$

$$\text{Comp}(x) \equiv (\exists y)[\text{Comp}(x, y)] \tag{29}$$

Similarly, Spec(x, y) holds iff x is a segment of a specifier of the maximal projection of y.

$$\begin{aligned}&\text{Spec}(x, y) \equiv \\ &\quad (\text{Base}(y) \vee \text{Bar2}(y)) \wedge \\ &\quad (\exists z_2)[\text{MaxProjection}(z_2, y) \wedge \text{Imm-Dominates}(z_2, x) \wedge \\ &\qquad \neg\text{Head-XP}(x, z_1)] \\ &\qquad\quad \text{—}x \textbf{ is the non-head child of the} \\ &\qquad\qquad \textbf{maximal projection of } y\end{aligned} \tag{30}$$

$$\text{Spec}(x) \equiv (\exists y)[\text{Spec}(x, y)] \tag{31}$$

10.4 Restricting Adjunction

Note that this definition of X-Bar structure allows both base and movement generated adjunction of anything anywhere, even at traces. We restrict this initially to allow only adjunction of heads to non-trace heads and XPs to non-trace XPs and $\overline{\mathrm{X}}$s.

$$(\forall x, y)[\mathrm{Adj}(x) \wedge y \triangleleft x \quad\rightarrow\quad \neg\mathrm{Trace}(y) \wedge \neg\mathrm{Lexicon}(y) \wedge (\mathrm{Bar0}(x) \wedge \mathrm{Bar0}(y) \vee \mathrm{Bar2}(x) \wedge \neg\mathrm{Bar0}(y)) \tag{32}$$

The predicate Trace, as with the predicate Base is defined in Section 13.

Base generated adjunction (that is where the adjoined element is not empty at D-Structure) can be detected by the conjunction of Adj and Base. Movement generated adjunction is characterized by Adj and ¬Base. Thus, one might, for instance, further restrict adjunction to prohibit base generated adjunction at X^0 (this is a common assumption):

$$\neg(\exists x, y)[\mathrm{Bar0}(x) \wedge x \triangleleft y \wedge \mathrm{Adj}(y) \wedge \mathrm{Base}(y)],$$

or of movement generated adjunction at A-positions (defined in the next section):

$$\neg(\exists x, y)[x \triangleleft y \wedge \mathrm{A\text{-}pos}(x) \wedge \mathrm{Adj}(y) \wedge \neg\mathrm{Base}(y)],$$

(adjunction of *wh*-phrases in these positions is barred in *Barriers*) and so on.

10.5 Argument Positions

As a matter of convenience, we will assume that argument positions can be identified by some reasonable, primarily structural, principles. The set we use here is taken from Fong 1991, pg. 218.

$$\begin{aligned}\mathrm{A\text{-}pos}(x) \equiv& \\ (\exists y)\ [&\mathrm{Lexical}(y) \wedge \mathrm{Theta\text{-}marks}(y) \wedge \mathrm{Comp}(x, y) \vee \\ &\mathrm{I}(y) \wedge \mathrm{Spec}(x, y) \vee \\ &\mathrm{N}(y) \wedge \mathrm{Spec}(x, y) \wedge \mathrm{N}(x)\quad]\end{aligned} \tag{33}$$

The first disjunct picks out, in Fong's words, "complement positions of lexical heads corresponding to internal θ-grid roles." This will be discussed more fully when we discuss Theta-theory in our definition of the lexicon (Chapter 11), for now we note that we assume that these are just the complements of lexical heads that assign Theta-roles. (Lexical(x) is true of the **lexical** categories—N, V, A, and P.) This assumption can be relaxed if the lexicon is modified such that Internal, say, is true of a node iff it is assigned an internal Theta-role. The second disjunct picks out [Spec,IP]. The third picks out [Spec,NP] if it is occupied by an NP (as opposed to a determiner). The details of the definition are not at all

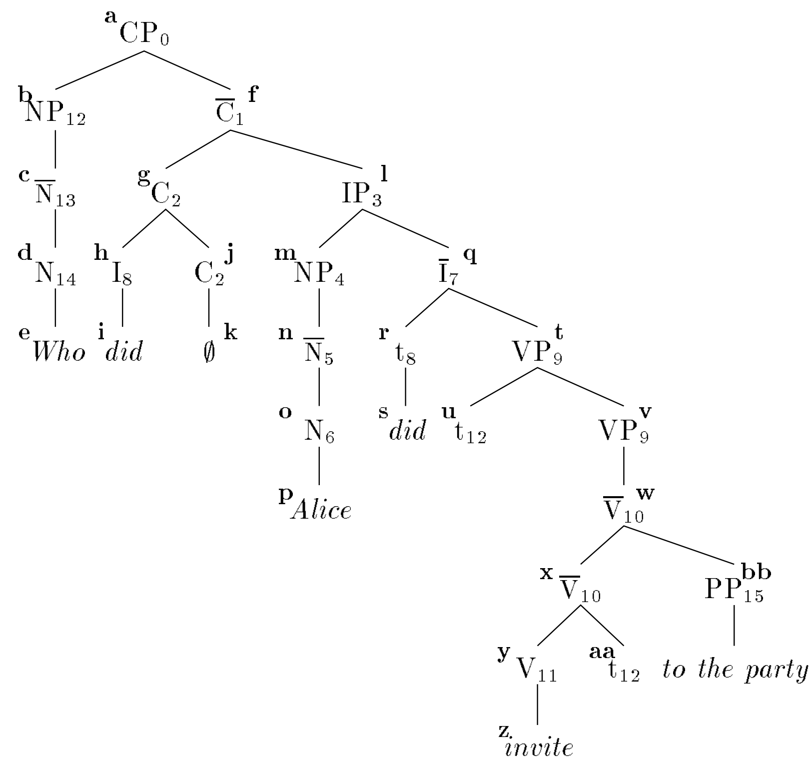

FIGURE 11 Structural Relations in X-Bar Theory

critical. The definition could be modified to accommodate a wide range of proposals governing the definition of A-positions.

10.6 An Example

Figure 11 illustrates the structural relationships we have defined. The predicate Category(X) is true of each of the sets $\{g, j\}$, $\{t, v\}$, and $\{w, x\}$, as well as singleton sets containing each of the nodes not in these three. Adj(x) is true of h, u, and bb and false for all others. MaxSeg(x) is false only for j, v, and x, while MinSeg(x) is false only for g, t, and w. All other nodes are both maximal and minimal in their category.

As an example of the domination, exclusion, etc., relationships consider these with respect to the category VP $= \{t, v\}$, (and thus, with respect to t and v individually as well). VP dominates every category (and node) in the subtree rooted at w, while it is dominated by those categories on the path between a and q. Every category that is a proper descendant of the nodes along that path are left-of VP and every category it includes. VP includes those categories that it dominates plus itself and u, that is, the categories in the subtree rooted at t. It excludes every other category in the tree—those categories that dominate or are left-of it. Finally, VP Imm-Dominates $\overline{\mathrm{V}}$ (w and x) but not u, since VP does not dominate u, and not bb, since bb is Adj.

The command relations have been illustrated with Figure 9. We will skip them here, and move on to the X-Bar relations. HeadXP and Head$\overline{\mathrm{X}}$ are true of $\langle a, f\rangle$, $\langle b, c\rangle$, $\langle l, q\rangle$, $\langle m, n\rangle$, ... and $\langle c, d\rangle$, $\langle f, g\rangle$, $\langle f, j\rangle$, $\langle n, o\rangle$, $\langle q, r\rangle$, ..., respectively. Note that Head$\overline{\mathrm{X}}$ is *not* true of $\langle f, h\rangle$, but *is* true of $\langle q, r\rangle$. Projects is true of the pairs $\langle a, f\rangle$, $\langle b, c\rangle$, $\langle c, d\rangle$, $\langle f, g\rangle$, $\langle j, k\rangle$, etc. (F.Eq is true of $\langle g, j\rangle$.) The Lexicon nodes are $e, i, k, p, s, z, \ldots$. Note that the trace of I_8 dominates a lexical item just as I_8 does. This is responsible for selecting the structure of the IP. We have Projects(r, s), Projects(q, r), Projects(l, q), and Projects(h, i), but of course $\neg$Projects(g, h). The bar levels of the non-lexicon nodes are indicated by their labels. Base is false for b, h, i, and u, and true for all others.

CP (a) is the Max-Projection for a, f, g, j, and k, but not for any category included by b or l, since these exclude CP, and not for h or i, since these are not in Base position. Similarly, Head for each of a, f, g, j, and k is the category $\{g, j\}$, Comp for these is l and Spec is b. For the IP, (nodes l, q, r, and s), Max-Projection is l, Head is r, Spec is m and Comp is the category VP $= \{t, v\}$, and thus, both segments t and v as well.

Finally, the argument positions are m and aa.

10.7 Head-Government

While c-command and m-command are the fundamental structural relationships in GB, they have unbounded scope whereas most of the theory is concerned with relationships between categories within the same or adjacent phrases. This limited range is captured in the notion of **government**. In Case Theory, for instance, a head assigns case only to those NPs that it governs. Similarly, in Binding Theory the relevant domain is the **governing category** of a nominal. Roughly, this is the minimal phrase containing both it and its governor. Existence of a governing head is also fundamental to the licensing of empty categories. Government appears, in GB, in two forms. These cases are all instances of **head-government**. The other form, *antecedent-government* will be treated, along with the licensing of empty categories, in our discussion of chains (Chapter 13).

Head-government is, to a large extent, m-command with restricted scope. While a category will usually have many m-commanding heads, the intent is that it have at most a single governing head. The scope of government is limited by two mechanisms: **barriers**—maximal projections that block government, and **minimality**—which requires a governor to be the closest possible governor. In the *Barriers* approach Chomsky 1986, most of the work is done by barriers (not surprisingly), and consequently, most XPs are barriers. The main exception is IP, which allows a verb to case-mark the subject of an infinitival complement in the case of *Exceptional Case Marking*, e.g.:

(1) Alice [$_{\mathrm{VP}}$believes [$_{\mathrm{IP}}$Bob to be a fool.]]

The analysis we adopt, Rizzi's *Relativized Minimality* (Rizzi 1990), in contrast to *Barriers*, determines the locality of government mostly on the basis of minimality (again not surprising). Here the idea is that no head can govern across another potentially governing head. Government into a complement phrase is normally blocked by the fact that the head of that phrase is a closer potential governor. It is possible in the case of ECM simply because, under the particulars of the definition of minimality the INFL does not intervene between its specifier and higher heads. Under this analysis, most XPs are not barriers, although barriers cannot be dispensed with entirely. In Rizzi's definition (following Cinque) (Rizzi 1990, pg. 112, note 6):

XP is a barrier if it is not selected by an X^0 not distinct from [+V].

where *not distinct from*[+V] means not [−V]. In our terms, barriers are XPs that are not complements of some head other than N^0 or P^0. For the most part, this comes down to specifier and adjunct XPs and complements of N^0 or P^0. Note, that, in contrast to the *Barriers* account, there is no notion of *barriers by inheritance*, the effect of these being subsumed by minimality.

$$\mathrm{Barrier}(x) \equiv \mathrm{Bar2}(x) \wedge \neg(\exists y)[\neg(\mathrm{N}(y) \vee \mathrm{P}(y)) \wedge \mathrm{Comp}(x, y)] \qquad (34)$$

Head-government, then, holds under m-command with no intervening barriers or potential head-governors (Rizzi 1990, pg. 25). Governors are lexical heads (V^0, N^0, A^0, or P^0), TENSE (T^0), and heads with non-empty agreement (+agr)—provided agreement actually holds between the governor and the governee. Heads with non-empty agreement, as a rule are finite INFLs and non-null AGRs (depending on which analysis is in force), but Rizzi also allows COMP to acquire +agr under some circumstances.

$$\mathrm{Head\text{-}Governs}(x, y) \equiv \qquad (35)$$

$$\begin{array}{l}\mathrm{Bar0}(x) \wedge \mathrm{Base}(x) \wedge \text{M-Command}(x,y) \wedge \\ (\mathrm{Lexical}(x) \vee \mathrm{T}(x) \vee +\mathrm{agr}(x) \wedge \mathrm{Agreement}(x,y)) \wedge \\ \neg(\exists z)[\mathrm{InterveningBarrier}(z,x,y)] \wedge \\ \neg(\exists z)[\mathrm{Bar0}(z) \wedge \mathrm{Intervenes}(z,x,y)]\end{array}$$

We have added the requirement that x be in base condition, as this seems to be necessary to prevent a finite INFL that has moved to COMP from head-governing its specifier (which is already head-governed by the trace of the INFL).

The predicate Agreement(x, y) holds just in case x and y agree on all relevant features. We will presume that this suffices to determine whether the head actually agrees with the potential governor. It is not inconceivable that some additional restrictions on the structural relationship between x and y may be needed in some cases, but the structural restrictions on Head-Governs as defined appear to suffice.

$$\begin{aligned}\mathrm{Agreement}(x,y) \quad \equiv \quad & (\mathrm{Sing}(x) \leftrightarrow \mathrm{Sing}(y)) \wedge \\ & (\mathrm{Plural}(x) \leftrightarrow \mathrm{Plural}(y)) \wedge \\ & (\mathrm{1st}(x) \leftrightarrow \mathrm{1st}(y)) \wedge \cdots \end{aligned} \tag{36}$$

An intervening barrier is just one that dominates the potential governee, but excludes the potential governor.

$$\begin{array}{l}\text{Intervening-Barrier}(z,x,y) \equiv \\ \qquad \mathrm{Barrier}(z) \wedge \mathrm{Dominates}(z,y) \wedge \mathrm{Excludes}(z,x)\end{array} \tag{37}$$

The predicate Intervenes(z, x, y) picks out a z that falls between, wrt command relations, x and y. Here, we are looking for intervening heads. When we define antecedent-government we will be concerned with either intervening XPs (for XP movement) or heads (for head movement). This is the *relativized* aspect of Rizzi's theory—minimality concerns only potential governors of the appropriate type. The structural condition of intervening is defined in terms of m-command for lexical heads and c-command for non-lexical heads. This is so non-lexical heads (as in I^0) do not block external government of their specifiers (Rizzi 1990, pg. 111, note 4).

$$\begin{array}{l}\mathrm{Intervenes}(z,x,y) \equiv \\ \qquad \text{C-Command}(z,y) \wedge \neg\text{C-Command}(z,x) \wedge \neg\mathrm{Lexical}(z) \quad \vee \\ \qquad \text{M-Command}(z,y) \wedge \neg\text{M-Command}(z,x) \wedge \mathrm{Lexical}(z)\end{array} \tag{38}$$

Figure 12 illustrates the definitions of barriers and head-government. Barriers are marked with rectangles while potential head-governors are

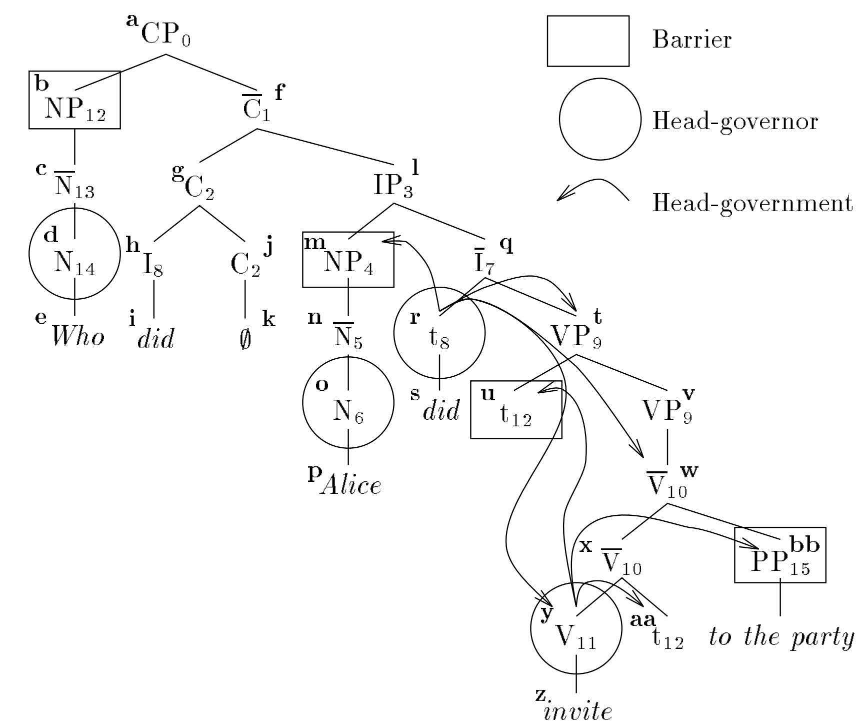

FIGURE 12 Barriers and Head-Government

marked with circles. Note that I_8 is not a potential head-governor since it is not in base position. It is, on the other hand, capable of blocking head-government from outside the CP, although this is irrelevant since such government is already blocked by C_2. Actual head-government is indicated by the arrows. Not every category is head-governed, nothing can govern IP_3, for instance, since C_2 blocks government from above but, being empty, does not itself head-govern. The nouns fail to head-govern since there is nothing that they m-command.

11

The Lexicon, Subcategorization, Theta-theory, and Case Theory

If the projection principle is taken literally, the entire syntactic mechanism is driven from the lexicon level. It is here that features are assigned, which then are transmitted, via projection, up through the phrase. This is also the level at which the selection of specifiers and complements is determined (by subcategorization requirements, etc.) as are the closely related mechanisms of Theta-marking and case assignment. Other components of the theory—Binding Theory, Control Theory, the theory of movement—are set in motion by the distribution of the categories and their features that are determined here. As we noted in the introduction, it has been observed elsewhere (Johnson 1989, Berwick 1991) that the properties assigned here all seem to involve local (in the technical sense) relationships. In a sense the treatment we sketch here serves mostly to confirm this observation. The actual details are unimportant; substantial variation is possible within our general approach.

There is great deal of regularity to the properties of the lexicon, but for the most part, we purposely avoid capturing this. Our intent is to maximize simplicity and generality, and so we usually avoid making even obvious assumptions about the distribution of properties across the lexicon. Most generalizations can be exploited either by adding disjunctions into our definitions (such as allowing for either NP or CP complements in a single entry for a word) or by explicitly extracting them from it (such as expressing $\mathrm{V}(x) \leftrightarrow +\mathrm{v}(x) \wedge -\mathrm{n}(x)$ as a separate principle, where V is the category and +v and −n are the usual $\pm V$ and $\pm N$ categorial features[1]). We assume that such generalizations have all

[1] These are ±v and ±n rather than just v and ¬n, etc., since not all categories share these features and one presumably wants to make generalizations over the negative features as well as the positive ones.

been multiplied out. We assume further that the result of this process is finite. Thus, we make no attempt to handle non-finite, let alone non-context-free, vocabularies, such as that of Bambara.

11.1 Principles Enforced in the Lexicon

As it is implemented here, a number of principles and parameters are either restrictions on or consequences of the definition of the lexicon. The **Extended Projection Principle**, for instance, requires, in part, that every sentence has a subject. The ***pro*-drop parameter** determines the availability of the empty subject *pro* to fulfill this requirement. Here these are fulfilled by requiring INFLs to select a specifier and by the presence or absence of *pro* in the lexicon. Agreement between subject and verb is also enforced in the definition of INFLs. Some aspects of the **Theta Criterion** and **case filter**, which together require that every chain containing an argument receives exactly one Theta-role and one case assignment, show up as requirements that every entry assigns a Theta-role to each argument it licenses and licenses an argument for each Theta-role it assigns, and that selectional constraints, case assignment features and the requirement that case assignment occur under head-government suffices to assign case uniquely and unambiguously.

The case filter itself, which requires that every overt NP receive case, is enforced by the fact that every overt NP in the lexicon has some case specified. Our definition insures that an NP with Accusative case, for instance, can only be licensed in positions marked with accusative case by some case assigner.

11.2 The Lexicon

The lexicon is realized as a large disjunction with a disjunct for each configuration of each word. As noted earlier, this includes entries for empty heads and for every possible COMP and INFL (or AGR and TENSE) including null ones.

$$
\begin{aligned}
\text{Lexicon}(x) \quad \equiv \quad & \text{see}(x) \wedge \text{V}(x) \wedge \cdots \quad \vee \\
& \text{seen}(x) \wedge \text{V}(x) \wedge \cdots \quad \vee \\
& \vdots \\
& \text{I}(x) \wedge \text{Finite}(x) \wedge \cdots \quad \vee \\
& \vdots
\end{aligned} \tag{1}
$$

The ellipses for each word includes specification of all appropriate linguistic features, explicit constraints on specifiers and complements, and explicit specification of the Theta-marking and case assigning properties of the word. So one might have, for *believe* as in *I believe the story*:

$$
\begin{aligned}
&\text{believe}(x)\wedge && (2a)\\
&\quad \mathrm{V}(x) \wedge +\mathrm{v}(x) \wedge -\mathrm{n}(x) \wedge \mathrm{Sing}(x) \wedge \mathrm{1st}(x) \wedge \mathrm{Finite}(x) \wedge \\
&\qquad \neg\mathrm{Passive}(x) \wedge \cdots && (2b)\\
&\quad \mathrm{Base}(x) \rightarrow ((\exists! y)[\mathrm{Comp}(y,x) \wedge \mathrm{MaxSeg}(y)] \wedge \\
&\qquad\qquad (\forall y)[\mathrm{Comp}(y,x) \rightarrow \mathrm{N}(y)] \wedge \\
&\qquad\qquad \neg(\exists y)[\mathrm{Spec}(y,x)]) \;\wedge && (2c)\\
&\quad \text{Marks-External-Agent}(x) \wedge \\
&\qquad \neg\text{Marks-External-Theme}(x) \wedge \neg \cdots && (2d)\\
&\quad \neg\text{Marks-Internal-1-Agent}(x) \wedge \\
&\qquad \text{Marks-Internal-1-Theme}(x) \wedge \ldots && (2e)\\
&\quad \text{Assigns-Acc}(x) \wedge \neg\text{Assigns-Nom}(x) \wedge \cdots && (2f)
\end{aligned}
$$

or for *believe* as in *Alice believes Bob to be asleep*:

$$
\begin{aligned}
&\text{believes}(x)\wedge && (3a)\\
&\quad \mathrm{V}(x) \wedge +\mathrm{v}(x) \wedge -\mathrm{n}(x) \wedge \mathrm{Sing}(x) \wedge \mathrm{3rd}(x) \wedge \mathrm{Finite}(x) \wedge \\
&\qquad \neg\mathrm{Passive}(x) \wedge \cdots && (3b)\\
&\quad \mathrm{Base}(x) \rightarrow ((\exists! y)[\mathrm{Comp}(y,x) \wedge \mathrm{MaxSeg}(y)] \wedge \\
&\qquad\qquad (\forall y)[\mathrm{Comp}(y,x) \rightarrow \mathrm{I}(y)] \wedge \\
&\qquad\qquad \neg(\exists y)[\mathrm{Spec}(y,x)]) \;\wedge && (3c)\\
&\quad \text{Marks-External-Agent}(x) \wedge \\
&\qquad \neg\text{Marks-External-Theme}(x) \wedge \neg \cdots && (3d)\\
&\quad \neg\text{Marks-Internal-1-Agent}(x) \wedge \\
&\qquad \text{Marks-Internal-1-Theme}(x) \wedge \ldots && (3e)\\
&\quad \text{Assigns-Acc}(x) \wedge \neg\text{Assigns-Nom}(x) \wedge \cdots && (3f)
\end{aligned}
$$

The second line (2b, 3b) of these entries define the basic linguistic features of the word. The third (2c, 3c) selects the structure of the phrase headed by x. These are qualified by Base(x) so they only apply to x in D-Structure position. Recall that our definition of X-Bar structure (Chapter 10) insures that, even in the case of head movement, there will always be a Lexicon item under the head of every phrase. In the case of the first sense of *believe* (2) there must be no specifier and a single NP complement. In the second sense (3) there must be no specifier and a single IP complement.

11.3 Theta-Marking

The fourth and fifth (2d–2e, 3d–2e) line of the lexicon entries determine the Theta-marking properties of the word. Their structure is intended to draw an analogy to Theta-grids. Theta-theory has to do with predicate-argument structure. Each predicate requires certain arguments—entities satisfying the relationship the predicate describes. In the case of *believe*, there must be an **Agent** (the believer) and a **Theme** (what the agent believes). The role assigned to an argument restricts the words that can be selected to occupy that position. Selection of *potato*, for instance, as the Agent of *believe* would be ruled out. There is little consensus on the exact range of these **Theta-roles** nor on the specific roles assigned to the arguments of some predicates, but there is generally agreement on the core cases. The **internal** Theta-roles are those realized by the complements (or perhaps adjuncts) of the predicate, for example the Theme of *believe*. Presumably, there may be any number of these, although two always seems to suffice and we will not bother to account for more than one here. The remaining roles are **external** roles. There is never more than one of these, and it is typified by the subject of a verb, the Agent of *believes*, for example.

Most commonly, the Theta-marking characteristics of a word are specified with a Theta-grid, a list with slots for each external and internal Theta-position which are filled with the appropriate role, if any. In our interpretation we have a feature for each position/role pair (Marks-External-Agent, Marks-External-Theme, Marks-Internal-1-Agent, etc.). Each lexicon entry, then, specifies at most one role for each position positively, and specifies all others negatively. The physical positions corresponding to the Theta-positions are assumed to be fixed. (Variation would be captured by expanding the set of positions.) Selectional restrictions are enforced by requiring the physical position corresponding to the Theta-position to be filled by a category bearing the Theta-role assigned to that position as a feature. So only XPs marked Theme (projected from the lexical head) can occupy the first internal Theta-position of *believe* in sense (2). Following our principle of avoiding generalizations in the lexicon, words that can fulfill multiple Theta-roles will be represented by multiple entries.

For simplicity we will assume that the internal arguments are simply the complements and are distinguished by their relative position. Its easy to see how even relatively complex conditions on this structure could be enforced. The predicate External-Arg picks out the external argument of a verb, that is, the specifier of the IP of which the verb is a complement.[2] Other external argument configurations, such as external arguments of nouns, can be treated similarly.

$$\begin{aligned}&\text{Internal-Arg-1}(x,y) \equiv \\ &\quad \text{Comp}(x,y) \wedge (\forall z)[\text{Comp}(z,y) \rightarrow \neg\text{Left-Of}(z,x)] \\ &\quad \text{—}x \text{ is the left-most complement of } y\end{aligned} \tag{4}$$

$$\begin{aligned}&\text{External-Arg}(x,y) \equiv \\ &\quad (\exists x_1, y_1)[\text{Max-Projection}(y_1,y) \wedge \text{Comp}(y_1,x_1) \wedge \text{Spec}(x,x_1)]\end{aligned} \tag{5}$$

With these, we can enforce selectional restrictions on the arguments

$$\begin{aligned}&(\forall x)[\text{Agent}(x) \leftrightarrow \\ &\quad (\exists y)[\ \text{Marks-External-Agent}(y) \wedge \text{External-Arg}(x,y) \vee \\ &\qquad\quad \text{Marks-Internal-1-Agent}(y) \wedge \text{Internal-Arg-1}(x,y) \vee \\ &\qquad\quad \vdots\end{aligned} \tag{6}$$

$$\begin{aligned}&(\forall x)[\text{Theme}(x) \leftrightarrow \text{etc.} \\ &\quad \vdots\end{aligned} \tag{7}$$

A category is Theta-marked, then, only if one of the Theta-roles has been required for its position.

$$(\forall x)[\text{Theta-Marked}(x) \leftrightarrow \text{Agent}(x) \vee \text{Theme}(x) \vee \cdots] \tag{8}$$

The **Theta criterion** requires that each argument receives exactly one Theta-role and each Theta-role is assigned to exactly one argument. For us, this reduces to a requirement that every argument position that is specified in the lexical entry for a word must correspond to exactly one positively specified Theta-position and *vice versa*.

11.4 Case Assignment

The final line of the lexicon entries (2f, 3f) determine the case-marking characteristics of the word. As we are working with English, we assume an impoverished case system, although it should be clear how this can be expanded to deal with both structural and inherent case, etc. As it stands, we treat only structural case and only **Nominative** and **Accusative** cases. We handle this much like our treatment of Theta marking. Since we consider only verbs with single complements, no head

[2] This is simpler if a VP-internal subject analysis is adopted, but the usual formulation of this analysis is inconsistent with Relativized Minimality. See Rizzi 1990, pg. 114.

assigns case to more than one NP. The entry simply specifies positively which case it assigns, if any.[3]

The relationship of the case assigner to the recipient of that case is not fixed to the same degree as the relationship between a predicate and its arguments. The first sense of *believe* simply assigns accusative case to its complement. The (3) sense, in contrast, is an *Exceptional Case Marking* verb, that is, it assigns accusative case to the subject of its infinitival complement. We specify, in both cases, only that the verb assigns accusative case. Case is always assigned under head-government, and always to **overt** NPs (NPs other than traces, PRO, or *pro*). Our assumption is that no case assigner head-governs more than one overt NP. Thus, we assume that the selection restrictions on specifiers and compliments along with the restricted domain of head-government suffice to unambiguously pick out the appropriate recipient of the case assignment. Again, case assignment, like Theta-role assignment, in our interpretation, is an agreement process. Every overt noun is assigned a case feature and the feature Overt in the lexicon. These can only be selected in positions that are marked with the same case by some case assigner.

$$(\forall x)[\mathrm{Acc}(x) \leftrightarrow \mathrm{N}(x) \wedge \mathrm{Overt}(x) \wedge (\exists y)[\text{Assigns-Acc}(y) \wedge \text{Head-Governs}(y,x)]] \tag{9}$$

$$(\forall x)[\mathrm{Nom}(x) \leftrightarrow \mathrm{N}(x) \wedge \mathrm{Overt}(x) \wedge (\exists y)[\text{Assigns-Nom}(y) \wedge \text{Head-Governs}(y,x)]] \tag{10}$$

$$\vdots$$

$$(\forall x)[\text{Case-Marked}(x) \leftrightarrow \mathrm{Acc}(x) \vee \mathrm{Nom}(x) \vee \cdots] \tag{11}$$

11.5 Other Lexicon Items

An entry similar to these first two would exist for the passive form of *believe*, as in *Alice is believed to be sleeping*. This would bear the feature Passive, and would be negatively specified for all external Theta-roles and for case-marking. This sense would be selected by an auxiliary, *was* for example. The entry for the auxiliary would require its complement to be marked Passive.

A finite, null INFL ($\emptyset$) with NP subject and VP complement might be:

$$\emptyset(x) \wedge \tag{12}$$

[3] This can be expanded to account for multiple case assignments by distinguishing multiple positions in the same way we do for Theta-marking.

$$
\begin{aligned}
&\mathrm{I}(x) \wedge \mathrm{Finite}(x) \wedge {+}\mathrm{agr}(x) \wedge \mathrm{Tense}(x) \wedge \mathrm{Sing}(x) \wedge \mathrm{1st}(x) \wedge \\
&\quad \mathrm{Past}(x) \wedge \cdots \\
&\mathrm{Base}(x) \rightarrow ((\exists! y)[\mathrm{Spec}(y,x) \wedge \mathrm{MaxSeg}(y)] \wedge \\
&\qquad\qquad (\forall y)[\mathrm{Spec}(y,x) \rightarrow (\mathrm{N}(y) \wedge \mathrm{Agreement}(x,y))] \wedge \\
&\qquad\qquad (\exists! y)[\mathrm{Comp}(y) \wedge \mathrm{MaxSeg}(y)] \wedge \\
&\qquad\qquad (\forall y)[\mathrm{Comp}(y) \rightarrow (\mathrm{V}(y) \wedge \mathrm{Agreement}(x,y))]) \ \wedge \\
&\neg\mathrm{Assigns\text{-}Acc}(x) \wedge \mathrm{Assigns\text{-}Nom}(x) \wedge \cdots
\end{aligned}
$$

The structure of this entry is similar to those of *believe*. Of course, INFLs assign no Theta-roles and so each Theta entry must be negatively specified, these have been omitted from the displayed portion of the entry. Finite INFLs assign Nominative case to the subject. Again, this is a consequence of Assigns-Nom and the fact that the INFL head-governs the subject. Subject-Verb agreement is enforced by requiring Agreement(x, y) to hold between both the subject and the verb and the INFL. This predicate requires agreement between the relevant features of x and y. It is defined in the discussion of head-government in Section 10.7 (Equation 36).

Note that the choice of a structure based on INFL and one based on AGR and TENSE is determined by the (expanded) lexicon.

12

Binding and Control

We turn now to two aspects of Government and Binding Theory that are normally treated in a way that depends heavily on co-indexation. These are the areas of Binding Theory and Control Theory. As we shall see, these are closely related topics, and our interpretation of Binding Theory will extend naturally to cover the relevant aspects of Control Theory. With some limitations, which are discussed below, we can capture Binding and Control Theory within $L^2_{K,P}$. Thus, this chapter serves as (the sketch of) a proof that indexation is not necessary to these theories (to the extent that we capture them) and that mechanisms to enforce them need not be more powerful than Context-Free Grammars. While these theories are reasonably amenable to treatment without indexation, our discussion will foreshadow issues that will arise in a more substantial form when we develop a treatment of the theory of movement without indexation in the next chapter.

We follow Haegeman (1991) in our exposition.

12.1 Binding Theory

Binding Theory concerns the interpretation of nominals, that is, of **reflexives**—*himself, yourself*, etc., and **reciprocals**—*each other*, etc., (collectively called **anaphors**), of **pronouns**—*him, she*, etc., and of **R-expressions** (referential expressions)—full NPs. This interpretation is usually indicated by co-indexation; NPs referring to the same entity are co-indexed. In

(1) Alice$_i$ thinks Bob$_j$ doesn't believe [she$_k$ saw him$_l$ herself$_k$]

Alice and *she* can either co-refer ($i = k$) or not ($i \neq k$), as can *Bob* and *him*. If $i = k$, then *Alice* resolves the reference of *she*—*Alice* **binds** *she*. An NP is bound iff there is c-commanding NP that is co-indexed with it—its **antecedent.** If there is no antecedent, in this case $i \neq k$, then the pronoun is said to be **unbound** or **free**. Binding, in this sense, is a structural relation requiring c-command and co-indexation. Since we are concerned with resolving references, and since these must be resolved (when they are resolved) by some argument that occurs in the sentence, we are only interested in binding by arguments—**A-binding**—also a structural relationship: c-commands and co-indexed from an A-position.

She and *herself*, in contrast to *Alice* and *she*, must co-refer. This is because anaphors must be bound locally:

(2) a. *He saw him herself.
    b. *She thinks [he saw herself].

In (2) no possible antecedent occurs sufficiently local to the anaphor, in (1) the only compatible antecedent is *she*.

Interpretation of pronouns, conversely, must not be local:

(3) a. *He$_i$ saw him$_i$.
    b. He$_i$ saw himself$_i$.
    c. He$_i$ thinks [he$_j$ saw him$_i$].
    d. He$_i$ thinks [he$_i$ saw him$_j$].

For R-expressions the restriction on interpretation is even stronger, these must be unbound everywhere in the sentence:

(4) a. *She$_i$ thinks [she$_j$ saw Alice$_j$].
    b. *She$_i$ thinks [she$_j$ saw Alice$_i$].

The nominals are categorized on the basis of these binding characteristics by two features, [±anaphor] and [±pronominal]:

Anaphors	[+anaphor, −pronominal]
Pronouns	[−anaphor, +pronominal]
R-expressions	[−anaphor, −pronominal]

The fourth category, [+anaphor, +pronominal] is associated with PRO, which is treated in the next section. The appropriate notion of sufficiently local is the **governing category** of the NP, the definition of which is rather delicate. It will be expanded, although only superficially motivated, shortly. With this, Binding Theory comes down to three Principles:

Principle A: An NP that is [+anaphor] must be bound in its governing category.

Principle B: An NP that is [+pronominal] must be free in its governing category.

Principle C: An NP that is [−anaphor, −pronominal] must be free everywhere.

It should be noted that there is still much that is unresolved within Binding Theory. There is controversy even over whether it applies at

S-structure, at LF, or both. While the trend seems to be towards its application at LF, since our interpretation focuses on S-structure we will follow a traditional approach (as rendered in Haegeman 1991) and apply it at S-structure. This should extend easily should we expand our target structures to incorporate LF.

There is a significant distinction between Principle A, on one hand, and Principles B and C on the other, in that Principle A requires the presence of an antecedent, while Principles B and C only prohibit certain patterns of co-reference. Sentences in which there are no potential antecedents of an anaphor in its governing category (Sentence (2) for instance) are ill-formed no matter how indices are assigned. For sentences that violate only Principles B and C, however, there will always be some assignment of indices—all distinct, at least—that will be grammatical. With our limited mechanism, we cannot, in general, impose or prohibit specific indexing. But, we can detect trees for which there is no acceptable indexing. That is to say, our structures do not determine co-reference, but we do not license any structure in which proper referents cannot be found. Consequently, Principles B and C have no direct effect on the set of trees we license.[1]

It is not the case, however, that Principles B and C have no effect at all. They do become part of a structural relationship Binding-Distinct(x, y) that holds between x and y only if x cannot bind y. As with all non-monadic relations, this must be explicitly defined. Thus, while it is useful in building definitions of other predicates (which can use it to check binding compatibility), it is actually a notational convenience and cannot be thought of as labeling the tree in the sense that the monadic predicates can. It is not available (except as a notational convenience), for instance, to mechanisms outside the one defined here that might check for proper indexation.

There is a sense in which indexation does not seem to have equal status with other aspects of syntactic structure. While the resolution of references is constrained by syntactic principles, it does not seem to be a purely syntactic property. This shows up in Chomsky 1993 (albeit in a context that argues, among other things, that Binding Theory cannot apply at S-structure, but rather must apply at LF) in the discussion of examples like:

(5) *Which claim that John$_i$ was a liar did he$_i$ deny?

In the theory developed there, the LF interpretation of this can take one of two forms:

[1] Except for the interaction of Principles A and B that partly determines the distribution of PRO.

(6) a. *[which x] [he$_i$ did deny x claim that John$_i$ was a liar]
b. [which x, x a claim that John$_i$ was a liar] [he$_i$ did deny x]

The judgment that the sentence, as indexed, is out is based on the Principle C violation that shows up in the first interpretation.[2] Chomsky argues that this interpretation prevails because it is preferred on the basis of economy. In general, failure to meet syntactic requirements can force the consideration of less economical derivations. Yet, the violation of Principle C in (6a) does not allow acceptance of the reading (6b). Principle C, then, is distinguished from most other syntactic principles in that it cannot apply before fixing on the economically preferred, but ill-formed interpretation.

12.1.1 Governing Categories

The governing category (GC) of an NP is, in its simplest form, the minimal IP or NP containing (as in Includes) it, its head-governor, and a c-commanding subject. In (1), the head-governor of *herself* is *saw* and *she* is a c-commanding subject. Thus, the IP *she saw him herself* is the GC for *herself.* If follows, then, that *herself* must be bound within this phrase by *she*, and can not be bound by an NP outside that phrase, *Alice*, for instance.[3] Unfortunately, this definition cannot account for judgments like:

(7) *Alice$_i$ thinks [$_{\mathrm{CP}}$ that [$_{\mathrm{IP}}$ herself$_i$ [$_{\mathrm{I}}$+agr] saw him]].

Here, there is no subject in the lower IP that c-commands *herself*, but the relevant GC is surely that IP. To account for these cases, the notion of SUBJECT is introduced, which is taken to include non-empty AGR. In the example in (7), the finite INFL of the lower IP is such a SUBJECT. The GC, then, must contain a c-commanding subject or an m-commanding SUBJECT.

But, even this is not quite sufficient. It fails to license, for instance:

(8) Alice$_i$ thinks [$_{\mathrm{CP}}$ that [$_{\mathrm{IP}}$ [$_{\mathrm{NP}}$ pictures of herself$_i$] [$_{\mathrm{I}}$+agr]
were seen by him.]]

The rationale applied here is that the SUBJECT—the finite INFL—is co-indexed, for agreement purposes, with [Spec,IP]—the NP *a picture of herself*, which dominates *herself.* If this SUBJECT were to be co-indexed with *herself*, it would violate the *i-within-i filter*, which forbids co-indexation of categories if one is dominated by the other. The proposal, then, is that a SUBJECT is **accessible** to a category only if co-

[2] Note that in the second interpretation there is no Binding Theory violation. *John$_i$* neither c-commands nor is c-commanded by *he$_i$*, and therefore, binding theory is not relevant.

[3] Unless *she* also binds *herself*, as in the case where $i = k$.

indexation of that SUBJECT and that category would not violate the i-within-i filter.

So, formally, a subject is [NP,IP] or [NP,NP]. A SUBJECT is a subject or a [+agr] head. These are accessible to a category if (for subjects) they c-command it, or, (for SUBJECTs) they m-command it and neither they nor their specifier violate the i-within-i filter. (Note that the c-command requirement for subjects subsumes the i-within-i filter.)

$$\mathrm{subject}(x) \equiv \mathrm{n}(x) \wedge (\exists y)[(\mathrm{i}(y) \vee \mathrm{n}(y)) \wedge \mathrm{Spec}(x,y)] \tag{1}$$

$$\mathrm{SUBJECT}(x) \equiv \mathrm{Subject}(x) \vee \mathrm{Bar0}(x) \wedge +\mathrm{agr}(x) \tag{2}$$

$$\begin{aligned} &\text{Accessible-Subject}(x,y) \equiv \\ &\quad \mathrm{subject}(x) \wedge \text{C-commands}(x,y) \quad \vee \\ &\qquad \text{---}x \textbf{ is a c-commanding subject} \\ &\quad \mathrm{SUBJECT}(x) \wedge \text{M-commands}(x,y) \wedge \\ &\qquad \text{--- } \textbf{or an m-commanding SUBJECT} \\ &\quad (\forall z)[(x \approx z \vee \mathrm{Spec}(z,x)) \rightarrow \neg\text{i-within-i}(z,y)] \\ &\qquad \text{---}\textbf{neither } x \textbf{ nor its specifier violate} \\ &\qquad\quad \textbf{i-within-i} \end{aligned} \tag{3}$$

The i-within-i filter, in its simplest form is just[4]

$$\text{i-within-i}(x,y) \equiv \mathrm{Dominates}(x,y). \tag{4}$$

The GC of x, then, is the minimal IP or NP including x, its head-governor, and a subject accessible to x. We define sets of nodes meeting the conditions first (gc), and then define the GC as the minimal such set of nodes. We define $\mathrm{GC}(y,x)$ to hold whenever y is in the GC of x.

$$\begin{aligned} \mathrm{gc}(X,x) \equiv\ & (\exists w,y,z)(\forall x') \\ & [(\mathrm{i}(w) \vee \mathrm{n}(w)) \wedge \mathrm{Bar2}(w) \wedge (X(x') \leftrightarrow \mathrm{Includes}(w,x')) \wedge \end{aligned} \tag{5}$$

[4] This is actually a little delicate. Haegeman (1991) cites the following example (due to Higginbotham) as evidence that i-within-i applies only to the configuration of an XP immediately dominating a category:

Mary$_i$ is [$_{\mathrm{NP}_i}$ [$_{\mathrm{NP}_j}$ [$_{\mathrm{NP}_i}$ her] cook's] best friend].

Following Haegeman's treatment (1991, pg. 227) we would have

$$\begin{aligned} \text{i-within-i}(x,y) \equiv\ & \mathrm{Dominates}(x,y) \wedge \\ & \neg(\exists z)[\mathrm{MaxProj}(z) \wedge \mathrm{Dominates}(x,z) \wedge \mathrm{Dominates}(z,y)] \end{aligned}$$

Unfortunately, this does not work for Example (8). We will leave open the issue of the precise definition of the i-within-i filter. Unless it is radically different than these, it will be within the capacity of $L^2_{K,P}$ in any case.

$$X(x) \wedge$$
$$X(y) \wedge \text{Head-Governs}(y, x) \wedge$$
$$X(z) \wedge \text{Accessible-Subject}(z, x) \quad]$$

$$\text{GC}(X, x) \equiv \text{gc}(X, x) \wedge (\forall Y)[\text{gc}(Y, x) \rightarrow \text{Superset}(Y, X)] \tag{6}$$

$$\text{GC}(y, x) \equiv (\exists X)[\text{GC}(X, x) \wedge X(y)] \tag{7}$$

We can now capture Principles B and C with a (necessarily) partial definition of the predicate Binding-Distinct(x, y). Note that this is an asymmetric relation. It is true only if x may not A-bind y. We don't presume to have a complete account of when this is the case, so we cannot hope to complete the definition. We do capture those that seem essential to the theory. In particular, x may bind y only if

- they agree in number, person, and gender,
- the i-within-i filter is respected,
- if y is [+pronominal] and x, in an A-position, c-commands y, then x is not in the GC of y (Principle B),
- if x is in an A-position c-commanding y, then y is not an R-expression ([−anaphor,−pronominal]) (Principle C).

$$\text{Binding-Distinct}(x, y) \equiv \tag{8}$$
$$\neg\text{Binding-Features-Agreement}(x, y) \quad \vee$$
$$\text{i-within-i}(x, y) \quad \vee$$
$$+\text{pronominal}(y) \wedge \text{A-pos}(x) \wedge \text{C-command}(x, y) \wedge \text{GC}(x, y) \quad \vee$$

—**Principle B violation**

$$-\text{anaphor}(y) \wedge -\text{pronominal}(y) \wedge \text{A-pos}(x) \wedge \text{C-command}(x, y) \quad \vee$$

—**Principle C violation**

$$\vdots$$

Binding-Feature-Agreement is just another agreement predicate, in this case enforcing number, person, and gender.

$$\text{Binding-Features-Agreement}(x, y) \equiv \tag{9}$$
$$(\text{Sing}(x) \leftrightarrow \text{Sing}(y)) \wedge \ldots \wedge$$
$$(\text{1st}(x) \leftrightarrow \text{1st}(y)) \wedge \ldots \wedge$$
$$(\text{Masc}(x) \leftrightarrow \text{Masc}(y)) \wedge \ldots$$

Principle A can then be expressed as a requirement that every [+anaphor] that has a GC has a potential antecedent in an argument position in that GC.

Principle A (10)

$$(\forall x, X)[+\text{anaphor}(x) \wedge \text{GC}(X, x) \rightarrow (\exists y)[X(y) \wedge \text{A-pos}(y) \wedge \text{C-command}(y, x) \wedge \neg\text{Binding-Distinct}(y, x)]]$$

It should be noted that this interpretation of Principle A does not identify a specific antecedent for a given anaphor; it only requires that such an antecedent exists. This is a fundamental characteristic of interpretations of relationships like these that do not employ indexation. Any number of anaphors might share the same potential antecedent if their GCs intersect. In English, at least as far as the definition of Binding-Distinct we have goes, this is not a problem. The only things that can force them to have independent reference is a clash of binding features or the *i*-within-*i* filter. But complex NP anaphors of the sort that could violate the *i*-within-*i* filter do not seem to occur in English. This comes down, then, to an issue of binding feature agreement. By the transitivity of agreement, the fact that the anaphors share an antecedent implies that they may co-refer.

12.2 Deriving the Distribution of PRO

The qualification that the anaphor has a GC figures in the standard account of the fact that PRO must be ungoverned. This is a topic for the next section, but as the argument is a Binding Theory argument we will anticipate that discussion and develop it here. The idea is that PRO is [+anaphor,+pronominal], and thus, subject to both Principles A and B. But one of these require it to be bound in its GC and the other requires it to be free in its GC. The resolution of the seeming contradiction is that PRO can have no GC, and, thus, must be ungoverned. This actually comes, as it should, as a consequence of our treatment of Binding Theory since

$$\begin{array}{ll} +\text{anaphor}(x), +\text{pronominal}(x) & \\ \Rightarrow \ \text{GC}(X, x) \rightarrow (\exists y)[X(y) \wedge \ldots \neg\text{Binding-Distinct}(y, x)] & \\ \qquad \textbf{(from (10))} & \\ \Rightarrow \ \text{GC}(X, x) \rightarrow (\exists y)[X(y) \wedge \ldots \neg\text{GC}(y, x)] & \\ \qquad \textbf{(from (8))} & \\ \Rightarrow \ \text{GC}(X, x) \rightarrow (\exists y)[X(y) \wedge (\forall Y)[\text{GC}(Y, x) \rightarrow \neg Y(y)]] & \\ \qquad \textbf{(from (7))} & \\ \Rightarrow \ \neg\text{GC}(X, x). & \end{array}$$

12.3 Control Theory

Control Theory concerns the resolution of the reference or PRO, the covert subject of otherwise subjectless non-finite IPs. For example:

(9) a. Alice$_i$ is wondering [$_{\text{CP}}$ whether [$_{\text{IP}}$ PRO$_i$ to invite Bob]].
b. Alice$_i$ is wondering [$_{\text{CP}}$ whether [$_{\text{IP}}$ PRO$_i$ to invite herself$_i$/*himself$_i$]].
c. [$_{\text{CP}}$ [$_{\text{IP}}$ PRO$_{arb}$ to invite Bob]] would be crazy.
d. *Alice$_i$ is wondering [$_{\text{CP}}$ whether [$_{\text{IP}}$ PRO$_i$ should invite Bob]].

That a subject occurs in the IP is required by the Extended Projection Principle. Clearly it is non-overt. In fact, it cannot be overt in these contexts since it is not head-governed,[5] and consequently, cannot receive case. This is characteristic of the distribution of PRO, it only occurs in positions that are ungoverned; hence, the ill-formedness of (9d).

In (9a) the controller of PRO is *Alice*. That PRO, in a context such as this, requires a controller can be seen in (9b). Here the reflexive requires a local binder. The only possibility is PRO. If the PRO could be freely interpreted, either reflexive would be acceptable. But the PRO is necessarily controlled by *Alice*, and so only the feminine reflexive works. In this context PRO behaves much like an anaphor. In (9c), in contrast there is no antecedent for PRO. It is said to be **arbitrary**. In this context it behaves more like a pronoun. This suggests the interpretation of PRO as [+anaphor, +pronominal] which, as argued in the previous section, accounts for its distribution only in ungoverned positions.

12.3.1 Obligatory and Optional Control

As we have seen, in some contexts PRO must be controlled. These are referred to as cases of **obligatory** control. A simple diagnostic of these cases is the unacceptability of PRO as a binder for the indefinite reflexive *oneself*:

(10) a. Alice decided [$_{\text{CP}}$ [$_{\text{IP}}$ PRO to invite herself/*oneself]]
b. Alice sent herself a note [$_{\text{CP}}$ [$_{\text{IP}}$ PRO to invite herself/*oneself]].

In other contexts, control is **optional**:

(11) a. John asked [$_{\text{CP}}$ how [$_{\text{IP}}$ PRO to behave himself/oneself]].[6]
b. [$_{\text{CP}}$ [$_{\text{IP}}$ PRO$_i$ to invite oneself$_i$]] would be rude.
c. [$_{\text{CP}}$ [$_{\text{IP}}$ PRO$_i$ to invite herself$_i$]] would be just like Alice$_i$.

[5] It is not head-governed because neither the non-finite INFL nor the COMP (which is empty here) are governors, but the COMP is a c-commanding head that intervenes between PRO and any other potential head-governor.

[6] This example is due to Manzini, via Haegeman (1991).

Whether control is obligatory or optional depends on the context in which PRO occurs. The examples above are typical. Control is obligatory for PRO occurring in declarative complement clauses (Sentence 10a) but not interrogative complement clauses (Sentence 11a). Neither is it obligatory in subject clauses (Sentences 11b and 11c), but it is obligatory in adjuncts (Sentence 10b). There are further refinements, but the point is that the nature of the control requirement, while somewhat idiosyncratic, depends on the structure, mood, etc. of the clause containing PRO or the clause containing that. We can capture essentially any condition that depends only on a bounded context like this. Thus, rather than go through the exercise of encoding this (partial) list, we will assume that the variants of PRO in the lexicon are of two varieties—those that are [+oblig] and those that are [−oblig]—and that the distribution of these is determined by their context.

12.3.2 Subject and Object Control

In some cases of obligatory control the controller of PRO must be the subject of the matrix clause, and in others it must be the object. Again, sentences in which the PRO binds a reflexive are diagnostic.

(12) a. Alice told Bob PRO to invite himself/*herself/*oneself.
b. Alice promised Bob PRO not to invite herself/*himself/*oneself.

Here, the variation is dependent on the verb of the matrix clause. Again, this is idiosyncratic. We will assume, then, that, in the lexicon, verbs of **subject control** will mark their subject [+controller] and those of **object control** will do the same with their object.

12.3.3 Control and Binding

Like Principles B and C, optional control can always be satisfied by indexing each NP distinctly. Thus, like those principles, optional control has no direct influence on the set of trees we license. In the case of obligatory control, as in Principle A, we must necessarily find a controller. This can rule out trees in two ways. There might be no argument in a controlling position, as in (from Haegeman 1991):

(13) *There occurred three more accidents without PRO being any medical help available on the premises.

In this example, the *There* in the subject position of the matrix clause is an expletive and cannot serve as an argument. The other configuration in which obligatory control rules out trees is the case where there is a controlling argument but it does not agree with the binding features of PRO. This can only occur when PRO binds another nominal necessarily (which thus, must be an anaphor) and has its features constrained by that relationship. This case accounts for the judgments we have been using throughout this section.

In cases of obligatory control, the controller must, evidently, c-command PRO.

(14) Alice's brother decided PRO to invite himself/*herself.

The controller cannot be *Alice*, which does not c-command PRO but can only be *Alice's brother*. (In cases of optional control, on the other hand, this is not a requirement—see Sentence (11c).) There are strong parallels, then between obligatory control and the binding of anaphors. Exploiting this, we treat obligatory control very much as we treat Principle A.

ObligatoryControl

$$\begin{aligned}(\forall x, X)[\quad & \mathrm{PRO}(x) \wedge \mathrm{Oblig}(x) \rightarrow \\ & (\exists y)[\text{C-command}(y,x) \wedge \mathrm{Controller}(y) \wedge \\ & \quad \neg\mathrm{Expletive}(x) \wedge \text{Binding-Features-Agreement}(y,x)]]\end{aligned} \tag{11}$$

We might have employed ¬Binding-Distinct here, but of the cases it covers, only the condition Binding-Features-Agreement is germane.

12.4 Discussion

In this chapter we have sketched definitions in $L^2_{K,P}$ of binding theory and control theory, or at least of those aspects of these theories that govern the distribution of nominals rather than their interpretation. The main result, of course, is that the principles of these aspects of these theories are, in some sense, context-free. In developing this result, though, other issues arise. We have been led, for instance, by our inability to employ indexation, to identify a distinction between the Principle A and Principles B and C of binding theory (and a similar distinction between cases of Obligatory and Optional Control). It is perhaps significant that this distinction is not idiosyncratic to our treatment, but has arisen in other analyses, from considerations of a more purely linguistic nature, as well.

Also, both as an example of the way in which formalizations of the theory support inferences, and as a partial verification of the formalization, we have sketched a formal derivation of the fact that PRO must be ungoverned (the PRO *theorem*) from its definition as being both anaphoric and pronominal and the principles of binding theory.

It should not be very surprising that these aspects of binding theory and control theory are definable in $L^2_{K,P}$. Obligatory Control is invariably resolved in the matrix clause and binding theory centers around the notions of head-government and governing category. These tend to bound the portion of the tree that is relevant to a given category with respect to the theories. In the next chapter we undertake a treatment of movement. This is a much more substantial task, since bounds such as these are harder to come by. Some of the issues that we have uncovered here will have central roles in that analysis, particularly the issue of identification without co-indexation and the problem of confusing the antecedents of categories.

13

Chains

The aspect of GB that is most difficult to capture in $L^2_{K,P}$ is the theory of movement. This, in fact, is where our interpretation fails, as indeed it must, for those non-English structures that have been shown to be non-Context-Free.

The role of indices in movement is to identify the moved category with its traces. Traces are the phonologically empty remnant of the category that marks the positions it visits in the course of movement. A moved element and its traces, collectively, form a **chain**. The D-structure position of the category is referred to as its **base** position. Each cycle (or step) of the movement forms a **link** in the chain. The new position is said to be the **antecedent** of the trace in the original position in the link. In the theory we adopt, all movement is assumed to be **raising** movement—to a higher position in the tree.[1] Thus, antecedents always c-command their traces, and, consequently, bind them (although not necessarily *A-bind* them). This also means that the link relation is reflected in the c-command relations among the members of the chain. One position in the chain is the antecedent of another just in the case that it is the minimal member of the chain that c-commands it. As a result, there is never any ambiguity about the sequence of moves that form a chain. The linear ordering of the chain imposed by the link relation is necessarily the same as its linear ordering by c-command. The maximal position must be the only position in the chain that is not a trace—the **target** of the movement. Consistent with the notion of binding, this position provides the identity of the entire chain.

As chains are identified with a single entity, they behave, in some

[1] This specifically eliminates the lowering I^0 to V^0 as a way for the verb to acquire inflection. We have already assumed, however, that verbs are fully inflected in the lexicon, and that verb morphology is only checked in the syntax rather than being generated there by movement.

respects like a single category. In particular, a chain can contain at most one argument (since it contains exactly one non-trace). That argument must receive exactly one Theta-role (by the *Theta Criterion*) and, if overt, must be marked with case (by the *Case Filter*). Chains formed by movement of overt NPs, then, must include exactly one position that receives a Theta-role and one that receives case. This is one of the factors that can force movement to occur.

In *Barriers* Chomsky 1986, the link relation is restricted by n-**subjacency**, a requirement that it cross no more than n barriers (n is a parameter of variation, but nearly always set at one.) Thus, it must hold in a bounded domain. This is attractive from our point of view, since the bounded domain of the link relation raises the possibility of a bound on the number of distinct links that can occur in the same (or overlapping) domains. In that case we can distinguish the links, and consequently, the chains that contain them, without resorting to (unbounded) indexation. Unfortunately, violations of subjacency are weak effects; there are many sentences in which they occur that, while degraded, are certainly not ungrammatical.

A stronger constraint on movement is the **Empty Category Principle** (ECP). In its traditional formulation, this is only partly a restriction on the link relation, and in many cases leaves it completely unbounded. One trend in accounts of movement is the idea that the structure of all chains should largely be determined by local conditions on links. Ungrammaticality that is usually attributed to ECP violations should then show up as ill-formed links. This again raises the possibility that we can capture all constraints on movement with principles that involve only a bounded portion of the tree. While this is not quite true—there are movements that involve links of arbitrary length (ignoring subjacency violations)—in English the number of such movements that can occur in a given sentence is bounded. Thus, the possibility that there is a bound on the number of links that can occur in overlapping domains is realized in English, and the interpretation we give here is reasonably complete. Structures, on the other hand, like the cross-serial dependencies of Swiss-German or long-distance extractions in Swedish, turn out to be cases in which the number of links that can interact is effectively unbounded.

The specific theory of link relations we implement is the core of Rizzi's *Relativized Minimality* theory. Following his lead, we will concentrate on ECP effects and ignore subjacency. We will pause here, before discussing Relativized Minimality, to characterize the class of movements we account for.

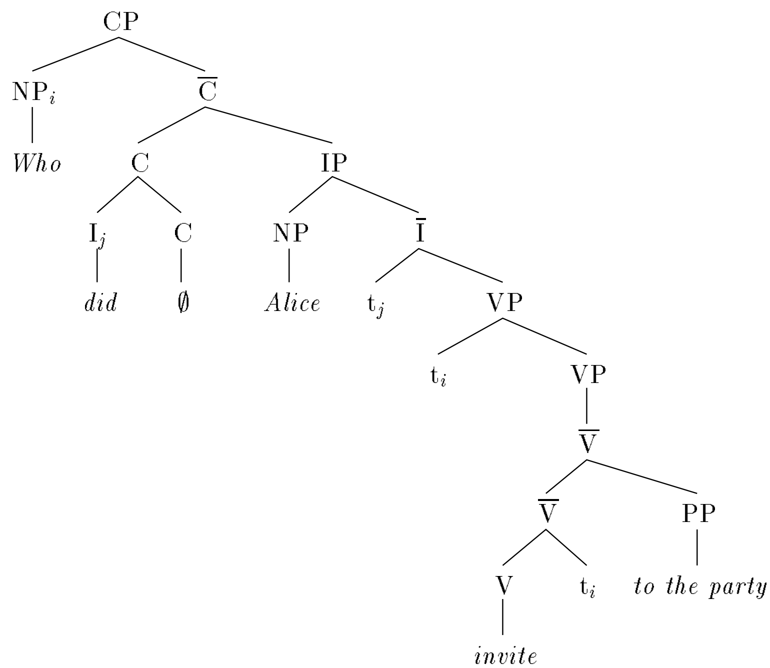

FIGURE 13 I-to-C Movement in English

13.1 A Taxonomy of Movement in English

This section is a brief, but reasonably complete, survey of the types of movement in English that have been widely studied. As with many aspects of GB theory, we cannot hope for this survey to be exhaustive, but it covers substantially all movement discussed in our sources, principally Radford 1988, Haegeman 1991, Rizzi 1990, Frank 1991, Manzini 1992, and, particularly for right movement, Kroch and Joshi 1987. Most of our examples are drawn from these sources.

The most basic distinction we make is on the basis of the Bar level of the moved category. Only X^0s and XPs move, so we distinguish two classes:

- **Head Movement** forms X^0-chains. In English, this is highly restricted, occurring only in cases of I^0 to C^0 and V^0 to I^0 to C^0 movement. As in

  (1) a. Who$_j$ did$_i$ [$_{\text{IP}}$ Alice t$_i$ invite t$_j$ to the party]?

  b. What$_j$ has$_i$ [$_{\text{IP}}$ Alice t$_i$ [$_{\text{VP}}$ t$_i$ [$_{\text{VP}}$ seen t$_j$]]]?

  The structure of Example (1a) is illustrated in Figure 13, that of Example (1b) is in Figure 14. (These are repeated from Chapters 8

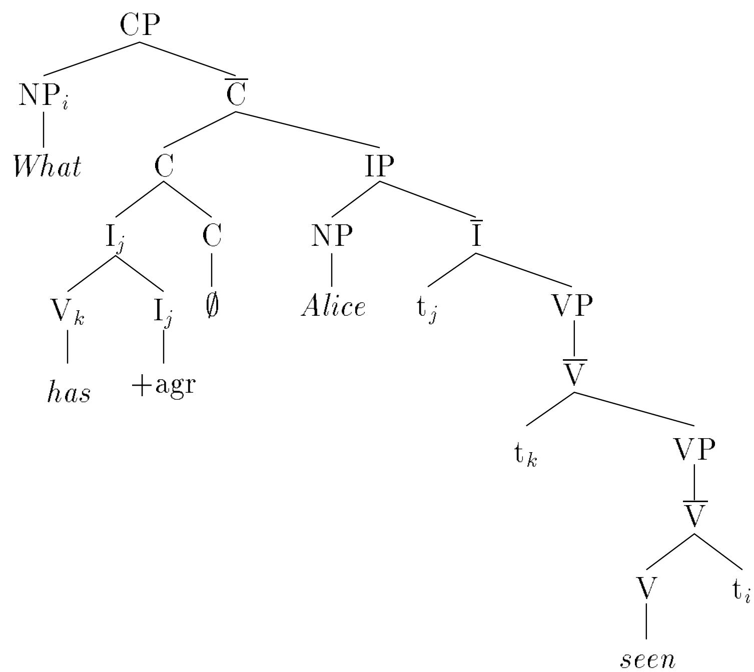

FIGURE 14 V-to-I-to-C Movement in English

and 10, respectively.) The chains of interest, of course, are those indexed i.

Head movement is often treated as either **amalgamation**—in which two heads are combined to produce a head that shares their characteristics—or substitution. As explained earlier, we treat all cases of head movement as adjunction. There is a universal constraint on head movement, the **Head Movement Constraint** which prohibits links in X^0-chains from crossing a head. Thus, the antecedent of an X^0-trace must be the closest c-commanding head.[2]

- **XP-movement** is the movement of maximal projections either by substitution of an XP at an appropriate position that was generated but left empty at D-structure, or by adjunction of an XP to some other maximal projection.

The class of XP-movements is refined further on the basis of the target position.

- **A-chains** are produced by movement from one argument position (A-position) to another.[3] Since there are no adjoined A-positions, this invariably involves substitution. The typical examples are:

  (2) a. Alice$_i$ was seen t$_i$ at the fair.

  b. Alice$_i$ seems [$_{\text{IP}}$ t$_i$ to have invited herself].

  c. *Alice$_i$ seems that it is likely t$_i$ to have invited herself.

  Sentence (2a) is the typical *passive* structure. Here *Alice*, the object of the verb, moves to an empty subject position as required by the passive *was seen*. Sentence (2b) is an example of *raising*. *Alice* is the subject of an embedded non-finite IP that is raised to the matrix subject position, again as a consequence of the nature of the verb.[4] The starred sentence, an example of *super-raising*, illustrates the strictly local nature of this type of movement—it cannot cross the subject position of the middle clause.

  Traces of A-movement are assigned features [+anaphor,−pronominal].

[2] While this would seem to provide exactly the kind of locality conditions we need, it turns out, in the case of cross-serial verb/object dependencies, that movements of multiple heads can interact to escape these bounds.

[3] This class of movement has traditionally been referred to as *NP-Movement*, although it need not involve only NPs.

[4] There is a superficial similarity between raising and control structures. In raising, at least in the account we adopt here, the embedded clause is necessarily an IP. Thus, the embedded subject position is governed by the matrix verb, and PRO cannot occur there. There are often similar forms in which the complement is a CP

It seems that Alice has invited herself.

but in these forms the embedded subject does not raise.

Thus, they behave like anaphors and, by Principle A, must be A-bound locally (by their antecedents). It follows that any intermediate traces must be in A-positions and the movement must, therefore, be by substitution. Movement in these cases is forced by the Case Filter. The chain receives a Theta-role at the base position, but is not case-marked there. Rather, the argument must move to a position in which it can receive case without receiving a second Theta-role. The only such position is the matrix subject, which receives case from the finite INFL, but is not assigned a Theta-role due to the nature of passive and raising verbs. A consequence of this analysis is that, since object positions always receive Theta-roles, there can be no A-movement into object positions. A-movement, then, is always movement of subject to subject or object to subject. For the class of movements we consider, this means that there is exactly one possible landing site, namely [Spec,IP].

- **$\overline{\text{A}}$-chains** involve movement of an XP into a non-argument position.[5] This can involve either substitution (into [Spec,CP], say) or adjunction to a maximal projection. Traces of $\overline{\text{A}}$-movement are assigned binding theory features [−anaphor,−pronominal]. Under Binding Theory (Principle C), these, like an R-expression, cannot be A-bound. Since they are necessarily bound by their antecedents, those antecedents may not occur in argument positions. Thus, A-chains necessarily involve A-positions, and $\overline{\text{A}}$-chains, *except for the base position* necessarily involve $\overline{\text{A}}$-positions. The exception allows these to intersect each other in exactly one way. An $\overline{\text{A}}$-chain and an A-chain can be concatenated in that order, e.g.:

  (3) Who$_i$ do you think [$_{\text{CP}}$ t$_i$ [$_{\text{IP}}$ t$_i$ is believed [$_{\text{IP}}$ t$_i$ to be winning]]].

  This can't be given a representational analysis as two chains, since it contains exactly one base position and one target position. If one views the essential aspect of chains to be the link relation, however, it has a simple analysis as a single chain containing both $\overline{\text{A}}$-links and A-links.

The class of $\overline{\text{A}}$-movement can be refined again into Left movement and Right movement based on the direction of movement in the surface string.

- **Right movement** Since all movement raises a category to a higher position in the tree, and since specifiers in English are ini-

[5] This is the class of movement traditionally referred to as *Wh-Movement*, although, again, it often involves categories other than *Wh* elements.

tial, right movement in English invariably involves adjunction to a c-commanding XP. This shows up in the surface string as movement of some constituent of a phrase either to the end of that phrase or to the end of a phrase containing that phrase.

(4) a. $[_{\mathrm{CP}}$ That $[_{\mathrm{IP}}$ $[_{\mathrm{NP}}$ someone $\mathrm{t}_i]$ will be there $[_{\mathrm{CP}_i}$ that you know]]] is likely.
b. $[_{\mathrm{CP}}$ That $[_{\mathrm{IP}}$ $[_{\mathrm{NP}}$ someone t_i] will be there $[_{\mathrm{PP}_i}$ from Peru]]] is likely.
c. Alice $[_{\mathrm{VP}}$ attended $[_{\mathrm{NP}}$ the reception $\mathrm{t}_i]$ last night $[_{\mathrm{PP}_i}$at Bob's house]].
d. *$[_{\mathrm{CP}}$ That $[_{\mathrm{IP}}$ $[_{\mathrm{NP}}$ someone t_i] will be there] is likely $[_{\mathrm{CP}_i}$ that you know]].

These are examples of extraposition of a CP (4a) and a PP (4b) from a subject, and a PP from an object (4c). As (4d) illustrates, this movement is strictly bounded by a condition similar to subjacency—the extraposed phrase cannot cross more than one major category. Thus, these are cases of movement from the complement or adjunct of an NP to adjoin at the minimal maximal projection including that NP.

(5) a. $[_{\mathrm{CP}}$ That $[_{\mathrm{IP}}$ Alice would $[_{\mathrm{VP}}$ show t_i to the guests $[_{\mathrm{NP}_i}$ the videos of her children]]]] was inevitable.
b. *$[_{\mathrm{CP}}$ That $[_{\mathrm{IP}}$ Alice would $[_{\mathrm{VP}}$ show t_i to the guests]]] was inevitable $[_{\mathrm{NP}_i}$ the videos of her children].

These are cases of heavy NP shift—an NP complement of the verb moves to adjoin to that VP. Again, as illustrated in (5b) the movement cannot cross more than one major category.

(6) a. Alice told Bob t_i yesterday $[_{\mathrm{CP}_i}$ that she wanted peaches].
b. Alice sent Bob t_i yesterday $[_{\mathrm{PP}_i}$ to the store to get some.]
c. How fond t_i are you $[_{\mathrm{PP}_i}$ of peaches].

Here we have rightward extraction of CP and PP arguments, (6a) and (6b) respectively, of a verb and a predicate adjective (6c). These pattern similarly to extraposition from NP.

The main characteristic of rightward movement in English, then, is that it involves movement of complements or adjuncts of a phrase to adjoin at the minimal XP properly including that phrase. It seems never to involve more than one cycle of movement. Thus, it is a strictly local form of movement.

- **Left movement** occurs in three general patterns:
  - Wh-question formation and Wh-exclamatives

    (7) a. $[_{\mathrm{CP}}$ Who_i $[_{\mathrm{IP}}$ t_i ate the peach]]?
    b. What_i did you eat t_i?

c. Why$_i$ did you eat the peach t$_i$?
d. [$_{\mathrm{AP}_i}$ How delicious] these peaches are.

(8) a. [$_{\mathrm{CP}}$ Who$_i$ did you think ...
[$_{\mathrm{CP}}$ t$_i$ [$_{\mathrm{IP}}$ t$_i$ ate the peach]]]?
b. What$_i$ did you think ... I ate t$_i$?
c. When$_i$ did you think ... I ate the peach t$_i$

These are cases of Wh-extraction from the subject, object and adjunct positions (Sentences (7a) through (7c), respectively). Wh-exclamatives (Sentence (7d)) are also a type of extraction, but being unaccompanied by inversion of the verb, do not form a question. The landing site, in each of these cases is [Spec,CP]. In general, the extraction can occur from an arbitrarily deeply embedded clausal complement, as shown in (8), although there are restrictions on the context from which they can be extracted that depend on the extraction site. (These will be covered shortly.)

- Topicalization and Preposing

(9) a. [$_{\mathrm{NP}_i}$The peach], I ate t$_i$ on purpose.
b. Peaches$_i$, we have plenty of t$_i$.
c. [$_{\mathrm{NP}_i}$The peach], I believe ... he ate t$_i$ on purpose.
d. Rarely$_i$ do you find such succulent peaches t$_i$.
e. She said she would eat one and [$_{\mathrm{CP}}$ [$_{\mathrm{VP}_i}$ eat one]
[$_{\mathrm{IP}}$ she did t$_i$]].
f. [$_{\mathrm{AP}_i}$ So hungry] would she have been t$_i$, that ...

These are examples of topicalization of a verbal complement (9a), a prepositional complement (9b), and a verbal complement of an embedded clause (9c), and preposing of an adjunct (9d), a VP (9e), and an AP complement (9f). Again, the landing site is [Spec,CP] in every case, and the extraction can generally be from a clausal complement.

- Relativization, Infinitival Adjuncts, Comparatives, and Parasitic Gaps

(10) a. [$_{\mathrm{NP}}$ The inconsiderate clod$_i$ [$_{\mathrm{CP}}$ O$_i$ that t$_i$ ate
my peach]]...
b. The peach$_i$ is too ripe [$_{\mathrm{CP}}$ O$_i$ [$_{\mathrm{IP}}$ PRO to eat t$_i$]].
c. The peaches are sweeter$_i$ than [$_{\mathrm{CP}}$ O$_i$ the pears
are t$_i$]
d. [Which fruit]$_i$ should you wash t$_i$
[$_{\mathrm{PP}}$ before [$_{\mathrm{CP}}$ O$_i$ [$_{\mathrm{IP}}$ PRO eating t$_i$]]].

These cases all involve a moved empty operator (O$_i$). In the case of parasitic gaps (10d), there is some controversy about the landing site. Chomsky, in *Barriers*, requires the empty operator to be licensed by a trace that is 0-subjacent. In the example, the licensing trace is t_i. If the operator is in [Spec,CP], as we have it, the PP will be a barrier intervening between the operator and that trace, violating this 0-subjacency condition. Consequently, he requires the operator to adjoin to the PP. Aoun and Clark (1985) propose an alternative analysis in which the empty operator is licensed by the operator in the matrix clause (*Which fruit*). Frank, in Frank 1991, critiques both of these approaches and offers an analysis of his own (based on TAGs) in which the empty operator is licensed by the operator in the matrix clause. Since neither of these require the 0-subjacency condition, the operator can land in [Spec,CP]. We will assume, then, an analysis that derives the structure we give in (10d). Then, once again, in each of the examples the movement lands in [Spec,CP], and originates in [Spec,IP], a complement or adjunct of VP or a complement or adjunct of a complement or adjunct of VP.

Left $\overline{\mathrm{A}}$-movement, then, is characterized by the fact that the target is always [Spec,CP] and the base is always either [Spec,IP] or is from a complement or adjunct that is more or less along the "spine" of the structure, or, more precisely is *not* from within a specifier.

Movement, in English at least, is characterized, then, by a few well-defined types, each with its own specific range of base positions and targets. Figure 15 displays these schematically. Our approach is to distinguish chains by type. We then need be concerned only with distinguishing chains of the same type with overlapping domains. Our claim will be that movement can be treated in such a way that the number of such overlapping chains is bounded, and thus, can be resolved without indexation.

13.2 The Empty Category Principle

As we noted in the previous sections there are a variety of phenomena that have been studied that reflect constraints on the context of the positions from which movement can occur. These include restrictions on head movement, *Subject-Object Asymmetries*, *that-Trace Effects*, *Wh-island Effects*, and many others. The usual account of these involves a combination of subjacency restrictions and the *Empty Category Principle.* Subjacency tends to be responsible for weak effects—questionable sentences rather than outright stars. The empty category principle is apparently much more fundamental, and, in Rizzi's analysis (Rizzi 1990), at least, it is the crucial constraint in the theory of movement.

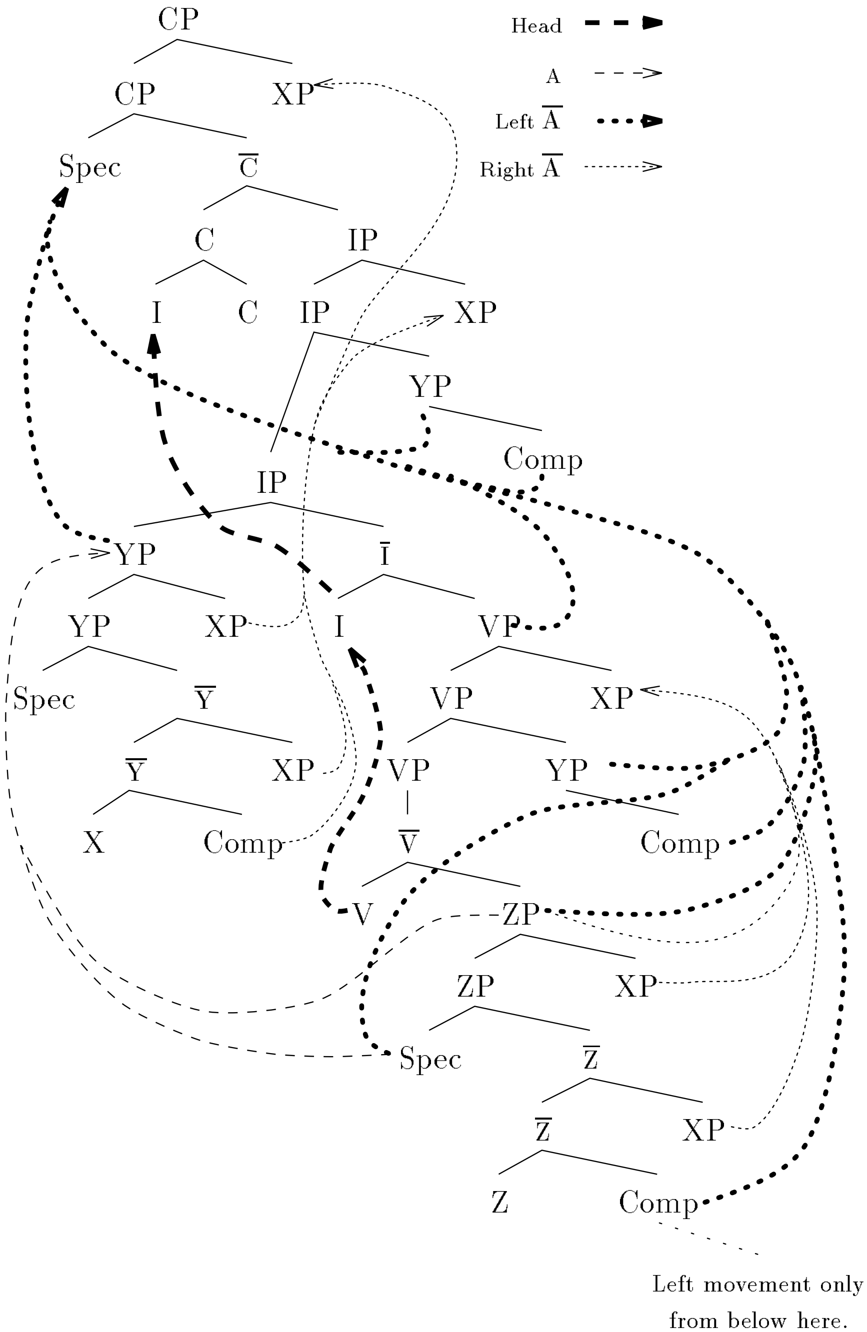

FIGURE 15 Movement in English

Empty Category Principle: Non-pronominal empty categories must be properly governed, where proper government is either *Antecedent-Government* or *Theta-Government*.

Essentially, this states that traces (non-pronominal empty categories) must be governed either by their antecedent or by the lexical head that assigns them a Theta-role. The alternatives show up in the account of subject-object asymmetries.

(11) a. Whom do you think Alice will invite t?
b. Whom do you think that Alice will invite t?
c. ?Whom do you wonder why Alice invited t?

(12) a. Who$_i$ do you think [$_{\mathrm{CP}}$t$_i$ [$_{\mathrm{IP}}$t$_i$ invited Alice]]?
b. *Who$_i$ do you think [$_{\mathrm{CP}}$t$_i$ that [$_{\mathrm{IP}}$t$_i$ invited Alice]]?
c. *Who do you wonder why t invited Alice?

The first set of sentences involves extraction from an object. Since objects are Theta-marked (and governed) by the verb, these satisfy the ECP regardless of the relationship between the trace and its antecedent. Sentence (11c) illustrates the fact that this relationship can still create subjacency effects.[6] In the second set of sentences, the extraction is from the subject. In (12a) both traces are antecedent-governed, but in (12b) government of the trace in [Spec,IP] is blocked by the overt complementizer *that*. (This is an example of the *that*-trace effect.)[7] Similarly, in (12c) the judgment is stronger than in (11c), since an ECP violation, and not just a subjacency violation, is involved.

Under Relativized Minimality, Rizzi isolates antecedent-government from head-government. Since minimality concerns only potential governors of the same type, potential head-governors (e.g., *that*) cannot block antecedent-government. To account for *that*-trace effects, then, he suggests a conjunctive form of the ECP involving two principles: formal licensing and identification.

ECP (Rizzi): A non-pronominal empty category must be both

- properly head-governed (Formal Licensing)
- antecedent-governed or Theta-governed (Identification).

The *that*-trace effect, in this analysis, is a failure of the formal licensing principle.

Rizzi goes on to question the disjunctive nature of the identification principle. He points out that there are phenomena mimicking subject-

[6] In this case the *why* prevents the movement from passing through the specifier of the lower clause, thus, forcing it to cross two barriers.

[7] The analysis of extraction from subjects is significantly different under *Relativized Minimality*.

object asymmetry that involve something beyond a simple subject-object distinction.[8]

(13) a. What did John weigh t?

b. ?What did John wonder how to weigh t?

In the first case *weigh* can be construed either in an agentive (*He weighed apples.*) or a stative (*He weighed 200 lbs.*) sense. But in the second, only the agentive is possible. He accounts for this by distinguishing **referential Theta-roles**, as assigned by the agentive sense of *weigh* from non-referential Theta-roles, as assigned by the stative. He argues that the only indices that are meaningful from the standpoint of identification are those assigned to arguments receiving referential Theta-roles. These he refers to as **Referential Indices**. He then categorizes $\overline{\mathrm{A}}$-chains as either **Referential Chains** or **Non-referential Chains**, on the basis of the nature of their index. This allows the disjunctive identification clause of the ECP to be subsumed under a general principle requiring every operator (the moved element) to be properly identified with its variable (the trace in the base position). For referential chains (including those of the standard cases of object extraction) this is satisfied by the referential index of the chain, but, since non-referential indices do not serve for identification, the principle can only be satisfied in non-referential chains (including cases of subject extraction) by the sequence of antecedent-government relationships linking the target of the chain to the base.

This might be stated:

ECP (Rizzi—final version):

- A non-pronominal empty category must be properly head-governed (Formal Licensing).
- Operators must be identified with their variables (Identification).

Identification: Operators are identified with their variables either by

- a referential index
- a chain of antecedent-government links.

In this way, Rizzi first introduces a conjunctive form of the ECP that extends the traditional disjunctive form, and subsequently replaces that disjunctive component with a distinction in the way that a single principle can be realized. From our point of view, of course, the most significant aspect of this analysis is that it reduces the role of indexation in chains to identification of referential chains, replacing it in non-referential chains with the local relationship of antecedent-government.

[8] Rizzi attributes the observation and the example to David Feldman.

13.3 Antecedent-Government

We can now proceed with formalizing these notions. Like that of head-government, the definition of antecedent-government has four components:

- The governor must be the proper type of category.
- The proper structural relationship must exist.
- There must be no intervening barriers.
- Minimality must be respected.

In the definition of head-government the appropriate structural relationship is m-command. For antecedent-government this is c-command. The notion of intervening barrier (which is weak in Rizzi's theory) remains unchanged. Since minimality is relative, it will be realized differently for each class of chain. Thus, there will be three versions of antecedent-government, one each for A-chains, $\overline{\text{A}}$-chains, and X^0-chains.

Usual definitions of antecedent-government make use of a co-indexation requirement to properly identify the antecedent. Since Rizzi has abandoned the use of indices for identification in non-referential chains, he suggests that this could be weakened to a general notion of non-distinctness; for referential chains this is interpreted as not having distinct indices, for non-referential chains antecedent-government must hold and, under minimality, the closest potential antecedent-governor must be the antecedent. Non-distinctness, then, simply rules out wildly incompatible antecedents (of the wrong category, etc.). We adopt a fairly strong interpretation of non-distinctness. We label each category in a chain with an additional complete set of features (we call these **T-Features**), that reflect the features of the target category. Thus, we can require that a trace and its antecedent agree in their T-Features. Note that the T-Features of chains formed by movement of identical categories will be identical. Thus, this is considerably weaker than co-indexation, as indeed it must be if it is to be definable, and so is weaker than Rizzi's notion of non-distinctness in the case of referential-chains. But in English, as we will see, other factors serve to eliminate any potential ambiguity in that case.

We add one more case of antecedent-government than Rizzi distinguishes. He does not treat right movement in English in *Relativized Minimality*, and it seems to be somewhat problematic for the theory. Rather than try to integrate it into $\overline{\text{A}}$-movement, we define Right-Antecedent-Governs as an independent case. This is a case where the Barriers that remain in Rizzi's theory apply effectively, and we define locality for this form of government with what is essentially a 1-subjacency condition.

$$
\begin{aligned}
&\text{A-Antecedent-Governs}(x, y) \equiv \\
&\quad \text{A-pos}(x) \wedge \text{C-Commands}(x, y) \wedge \text{T.Eq}(x, y) \wedge \\
&\qquad \text{—}x\ \textbf{is a potential antecedent in an A-position} \\
&\quad \neg(\exists z)[\text{Intervening-Barrier}(z, x, y)] \wedge \\
&\qquad \textbf{—no barrier intervenes} \\
&\quad \neg(\exists z)[\text{Spec}(z) \wedge \text{A-pos}(z) \wedge \text{C-Commands}(z, x) \wedge \\
&\qquad \text{Intervenes}(z, x, y)] \\
&\qquad \textbf{—minimality is respected}
\end{aligned} \tag{1}
$$

$$
\begin{aligned}
&\overline{\text{A}}\text{-Antecedent-Governs}(x, y) \equiv \\
&\quad \neg\text{A-pos}(x) \wedge \text{C-Commands}(x, y) \wedge \text{T.Eq}(x, y) \wedge \\
&\qquad \text{—}x\ \textbf{is a potential antecedent in an } \overline{\text{A}}\textbf{-position} \\
&\quad \neg(\exists z)[\text{Intervening-Barrier}(z, x, y)] \wedge \\
&\qquad \textbf{—no barrier intervenes} \\
&\quad \neg(\exists z)[\text{Spec}(z) \wedge \neg\text{A-pos}(z) \wedge \text{C-Commands}(z, x) \wedge \\
&\qquad \text{Intervenes}(z, x, y)] \\
&\qquad \textbf{—minimality is respected}
\end{aligned} \tag{2}
$$

$$
\begin{aligned}
&\text{Right-Antecedent-Governs}(x, y) \equiv \\
&\quad \text{Adj}(x) \wedge \text{C-Commands}(x, y) \wedge \text{T.Eq}(x, y) \wedge \\
&\qquad \text{—}x\ \textbf{is a potential antecedent in an adjoined position} \\
&\quad \neg(\exists z, z')[\text{Intervening-Barrier}(z, x, z') \wedge \\
&\qquad\quad \text{Intervening-Barrier}(z', z, y)] \\
&\qquad \textbf{—no more than one barrier intervenes}
\end{aligned} \tag{3}
$$

$$
\begin{aligned}
&\text{X}^0\text{-Antecedent-Governs}(x, y) \equiv \\
&\quad \text{Bar0}(x) \wedge \text{C-Commands}(x, y) \wedge \text{T.Eq}(x, y) \wedge \\
&\qquad \text{—}x\ \textbf{is a potential } \overline{\text{X}}\textbf{ antecedent} \\
&\quad \neg(\exists z)[\text{Intervening-Barrier}(z, x, y)] \wedge \\
&\qquad \textbf{—no barrier intervenes} \\
&\quad \neg(\exists z)[\text{Bar0}(z) \wedge \text{C-Commands}(z, y) \wedge \text{Intervenes}(z, x, y)] \\
&\qquad \textbf{—minimality is respected}
\end{aligned} \tag{4}
$$

Intervening-Barrier and Intervenes are defined with head-government in section 10.7 (Equations 37 and 38). The predicate T.Eq is just another agreement predicate forcing the T-features of the categories to be equal. We realize these T-features as a set of features T.N, T.V, ..., T.Sing,

..., one for each feature that can appear on a category.

$$\text{T.Eq}(x, y) \equiv (\text{T.N}(x) \leftrightarrow \text{T.N}(y)) \wedge (\text{T.V}(x) \leftrightarrow \text{T.V}(y)) \wedge \cdots \qquad (5)$$

13.4 The Link Relation

In Rizzi's analysis, chains are just sequences of antecedent-government links. Their primary function is the identification of the target and base positions. Technically, this would leave the case of long-distance referential movement outside the theory of chains. This actually becomes the basis of his analysis of the fact that long-distance A-movement is not possible even though the moved category typically receives a referential Theta-role. The Theta criterion, recall, expressed in terms of chains, requires every argument to belong to a chain that receives exactly one Theta-role, and every Theta-position to belong to a chain that includes exactly one argument. A-movement is generally forced by the case filter—the NP does not receive case in its base position, and so must move to a position in which it does. This can be incorporated into the Theta Criterion by requiring a chain to be marked with case in order to receive a Theta-role (the so called *visibility condition*). In $\overline{\text{A}}$-movement, both case and the Theta-role are assigned at the base position. Thus, the Theta Criterion is satisfied even if no (non-trivial) chain exists. In contrast, A-chains receive their Theta-role at the base position and their case at the target. Thus, the chain is required in order to satisfy the Theta Criterion.

This leaves $\overline{\text{A}}$-movement of referential objects as the sole exception to the antecedent-government requirement. Rather than leaving this type of movement outside the theory of chains, we choose to make this exception in the definition of a link. For all other forms of movement the link relation is just the appropriate notion of antecedent-government (coupled with corresponding restrictions on the types of the categories forming the link). For $\overline{\text{A}}$-referential-movement we will use a weakened notion, $\overline{\text{A}}$-Antecedent, in which intervening barriers may occur and minimality need not be satisfied.

We will assume that base positions receiving a referential Theta-role will be marked, by the head assigning that role, with the feature Ref. Propagation of this along the chain is then forced by the definition of $\overline{\text{A}}$-Ref-Link.

$$\begin{aligned} &\overline{\text{A}}\text{-Antecedent}(x, y) \equiv && (6)\\ &\quad \neg\text{Apos}(x) \wedge \text{C-Commands}(x, y) \wedge \text{T.Eq}(x, y) \\ &\qquad \text{—}x \textbf{ is a potential antecedent in an} \end{aligned}$$

$\overline{\mathrm{A}}$-position

We now have five types of links.

$$\text{A-Link}(x, y) \equiv \tag{7}$$

$\text{A-Antecedent-Governs}(x, y) \wedge$

$\neg\text{Base}(x) \wedge \text{Trace}(y){+}\text{anaphor}(y) \wedge -\text{pronominal}(y)$

—y **is an A-trace,** x **is not in Base position**

$$\overline{\text{A}}\text{-}\overline{\text{Ref}}\text{-Link}(x, y) \equiv \tag{8}$$

$\overline{\text{A}}\text{-Antecedent-Governs}(x, y) \wedge \neg\text{Ref}(x) \wedge \neg\text{Ref}(y) \wedge$

$\text{Bar2}(x) \wedge (\neg\text{Target}(x) \vee \text{Spec}(x)) \wedge$

—x **is an XP and is a specifier if it is the target**

$\neg\text{Base}(x) \wedge \text{Trace}(y) \wedge -\text{anaphor}(y) \wedge -\text{pronominal}(y)$

—y **is an** $\overline{\text{A}}$**-trace,** x **is not in Base position**

$$\overline{\text{A}}\text{-Ref-Link}(x, y) \equiv \tag{9}$$

$\overline{\text{A}}\text{-Antecedent}(x, y) \wedge \text{Ref}(x) \wedge \text{Ref}(y) \wedge$

$\text{Bar2}(x) \wedge (\neg\text{Target}(x) \vee \text{Spec}(x)) \wedge$

—x **is an XP and is a specifier if it is the target**

$\neg\text{Base}(x) \wedge \text{Trace}(y) \wedge -\text{anaphor}(y) \wedge -\text{pronominal}(y)$

—y **is an** $\overline{\text{A}}$**-trace,** x **is not in Base position**

$$\text{Right-Link}(x, y) \equiv \tag{10}$$

$\text{Right-Antecedent-Governs}(x, y) \wedge$

$\neg\text{Base}(x) \wedge \text{Trace}(y) \wedge -\text{anaphor}(y) \wedge -\text{pronominal}(y) \wedge$

—y **is an** $\overline{\text{A}}$**-trace,** x **is not in Base position**

$\text{Bar2}(x) \wedge \text{Target}(x)$

—x **is the Target and is an XP**

$$\text{X}^0\text{-Link}(x, y) \equiv \tag{11}$$

$\text{X}^0\text{-Antecedent-Governs}(x, y) \wedge$

$\neg\text{Base}(x) \wedge \text{Trace}(y)$

—x **is not in base position,** y **is a trace**

$$\begin{aligned}\text{Link}(x, y) \quad \equiv \quad & \text{A-Link}(x, y) \vee \overline{\text{A}}\text{-}\overline{\text{Ref}}\text{-Link}(x, y) \vee \\ & \overline{\text{A}}\text{-Ref-Link}(x, y) \vee \text{X}^0\text{-Link}(x, y) \vee \\ & \text{Right-Link}(x, y)\end{aligned} \tag{12}$$

The basic link relations are pairwise mutually exclusive. To see this, note that $\text{X}^0\text{-Link}(x, y)$ requires x to be at Bar0, while in each of the others it is required to be an XP. (In the case of A-Antecedent-Governs this is a consequence of A-pos(x).) The fact that x is required to be in an A-position distinguishes A-Link(x, y) from the other XP links, which require x to be in an $\overline{\text{A}}$-position. Right-Link(x, y) is distinguished from the other $\overline{\text{A}}$-links by the fact that it requires x to be a Target (right movement is non-cyclic in our analysis) and adjoined (which implies that it is not a specifier), while the others require x to be a specifier if it is a Target. Finally, the two types of $\overline{\text{A}}$-leftward movement are distinguished by Ref.

13.4.1 Avoiding Conflation of Chains

Our idea is to define chains as sets of categories that are linearly ordered by the link relation and contain both a maximum (the target position) and minimum (the base position) wrt to that relation. The remaining concern is that it is not sufficient to just pick out chains to satisfy ECP for the various traces in the tree, we must also insure that the chains are identified consistently for all traces in the tree. With indexation, this is simple. Each category belongs to exactly one chain, that consisting of all categories sharing its index. Here we have the possibility that two chains that occur within the same domain might end up intersecting. An example might be:

(14) * Who has [$_{\text{IP}}$ t$_i$ told you [$_{\text{CP}}$ t$_j$ [$_{\text{IP}}$ t$_j$ invited him]]].

where the indices on the traces are intended to distinguish them for discussion only. While chains can be found for both the i and j traces, they necessarily share *Who*. This can be read of a conflation of the sentences

(15) Who$_j$ has Alice told you [$_{\text{CP}}$ t$_j$ [$_{\text{IP}}$ t$_j$ invited him]]
Who$_i$ has t$_i$ told you Alice invited him.

In order to rule out such configurations, we define chains not only to be bounded sets of categories that are linearly ordered by the link relation, but also to be maximal in the sense that every category that is related to some member of the chain by the link relation is included in the chain. In the case of (14) any such set including either of the traces will necessarily include both, since both are related to *Who* by link. Consequently, there is no linearly ordered set including either of them that is maximal in this sense, and no chains for them can be found. This approach requires, though, that whenever two chains overlap in a well-formed sentence, there is no category in one that is related by the link relation to a category in the other. Otherwise, neither chain could be licensed. We argue, in the next section, that the classification of chains we have already developed, with some extensions, suffices to establish this condition for English.

13.5 Defining Chains

Chains, then, are bounded sets of categories that are linearly ordered by the link relation and are maximal in the sense of the preceding section.

$$
\begin{aligned}
&\text{Chain}(X) \equiv \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (13)\\
&\quad (\exists! x)[X(x) \wedge \text{Target}(x)] \wedge (\exists! x)[X(x) \wedge \text{Base}(x)] \wedge\\
&\qquad\quad \text{—}X \textbf{ contains exactly one Target and one Base}\\
&\quad (\forall x)[X(x) \wedge \neg\text{Target}(x) \rightarrow (\exists! y)[X(y) \wedge \text{Link}(y,x)]] \quad \wedge\\
&\qquad\quad \textbf{—All non-Target have a unique antecedent in } X\\
&\quad (\forall x)[X(x) \wedge \neg\text{Base}(x) \rightarrow (\exists! y)[X(y) \wedge \text{Link}(x,y)]] \quad \wedge\\
&\qquad\quad \textbf{—All non-Base have a unique successor in } X\\
&\quad (\forall x,y)[X(x) \wedge (\text{Link}(x,y) \vee \text{Link}(y,x)) \rightarrow X(y)]\\
&\qquad\quad \text{—}X \textbf{ is maximal wrt the Link relation}
\end{aligned}
$$

The maximality condition is enforced by the last clause. Note that $\text{Link}(x, y)$ cannot hold if x is Base or y is Target (under the assumption that $\text{Target}(y) \rightarrow \neg\text{Trace}(y)$). Thus, the Target position of the chain is necessarily maximal and the Base position necessarily minimal with respect to Link. It also follows that chains that would be conflated if they were to overlap can occur end-to-end without interfering with each other.

There is one more detail we need to take care of which has to do with the propagation of features along chains. The definition of Link requires a trace and its antecedent to agree in their T-features, but nothing yet requires the T-Features to take any particular value. Further, in movement some of the features of the moved category are inherited by its trace, usually the ϕ features—number, case, gender, etc.—but we can be neutral about which features they actually are. They do not, of course, include the features distinguishing binding properties ($\pm$anaphor and $\pm$pronominal) since these are assigned to traces on the basis of the type of movement and are independent of the type of the moved category. Both of these issues have to do with the relation between the T-features and the ordinary features of a category. We handle this as explicit principles, one which applies to targets and one which applies to traces:

$$
\begin{aligned}
(\forall x)[\quad & \text{Target}(x) \rightarrow \text{T.N}(x) \leftrightarrow \text{N}(x) \wedge \cdots \qquad\qquad (14)\\
& \wedge \text{T.+anaphor}(x) \leftrightarrow \text{+anaphor}(x) \wedge \cdots]\\[1em]
(\forall x)[\quad & \text{Trace}(x) \rightarrow \text{T.N}(x) \leftrightarrow \text{N}(x) \wedge \cdots]
\end{aligned}
$$

13.6 Defining ECP

We are now ready to return the the Empty Category Principle. In Rizzi's final version, this comes down to two principles: every trace must be properly head-governed, and every moved category must be identified with its base position. The second of these, which is the remnant of the traditional formulation of ECP, requires the trace to be either bound by its antecedent via a referential index, or to be antecedent-governed. By incorporating $\overline{\mathrm{A}}$-referential links into our account of chains, we reduce both of these cases to a requirement that the trace occurs in a well-formed chain. Thus, in our interpretation, the traditional notion of ECP reduces to the intuitively obvious requirement that every trace occurs in some chain.[9]

Proper head-government is just head-government of a category that is included in the minimal category including the governor.

$$\begin{aligned}&\text{Proper-Head-Governs}(x,y) \equiv \\ &\qquad \text{Head-Governs}(x,y) \wedge (\forall z)[\text{Includes}(z,x) \rightarrow \text{Includes}(z,y)]\end{aligned} \tag{15}$$

ECP, then, is expressed by the two principles:

Licensing (16)

$$(\forall x)[\text{Trace}(x) \rightarrow (\text{Bar0}(x) \vee (\exists y)[\text{Proper-Head-Governs}(y,x)])]$$

Identification (17)

$$(\forall x)[\text{Trace}(x) \rightarrow (\exists X)[\text{Chain}(X) \wedge X(x)]]$$

Rizzi requires only antecedent-government for X^0 movement (Rizzi 1990, pg. 118, note 8). As antecedent-government is required by the Identification principle, we exclude X^0 traces from the Licensing clause.

Of course, the requirement that Traces occur within some chain applies to Targets as well, that is, every category that is $\neg$Base should be assigned to some chain. But we can generalize this even further. As we admit trivial chains (in which the only member is both Target and Base), *every* category belongs to some chain. Then the Identification principle applies universally:

[9] The cost, of course, is that our notion of well-formed chain is much more complex. We have transferred the burden usually carried by ECP to the definition of chain. But without indexation this additional complexity in the definition of chains seems to be necessary in any case. It is satisfying, then, that it can be accompanied by a corresponding simplification of ECP.

Identification(Generalized) (18)

$$(\forall x)(\exists X)[\mathrm{Chain}(X) \wedge X(x)]$$

13.7 Distinguishing Chains in English

To complete this account of movement we still need to justify our claim that only ill-formed chains will be excluded by our maximality condition. In essence, this is a claim that we can distinguish a set of classes of chains, the size of which is bounded independent of the input, and which suffices to guarantee in well-formed English sentences that chains within the same class will never overlap. We will approach this by treating each of the classes of chains we have already distinguished separately. As we shall see, it is only in the case of rightward movement that we will have to refine any of these classes. As an aid in following the discussion, the reader will probably want to refer back to our map of the classes we cover in our taxonomy of movement in English, Figure 15.

We should note that our entire analysis up to this point ignores the fact that traces and targets themselves are sometimes moved in the course of the movement of other categories. This last issue is usually resolved by reconstruction which we will take in the next section.

13.7.1 Head Movement

In English, head movement is extraordinarily simple, particularly if verbs are fully inflected in the lexicon, as we have assumed. It comes down simply to I^0-to-C^0 or V^0-to-I^0-to-C^0 movement. Each movement is to the immediately c-commanding head and it never crosses a CP. In any domain there is a fixed bound on the number of categories that can participate in this movement and each is distinct from the others. Thus, while these chains can overlap (in the case of V^0-to-I^0-to-C^0 movement the V^0 chain and I^0 chain overlap) they are always distinguished by their T-features. In fact, even if they were not distinguished by their T-features, as long as there is a bound on the number of chains that can overlap we can always add a (bounded) set of new features sufficient to distinguish them. The inability of this approach to account for cross-serial dependencies in Dutch is a result of the fact that, in the analysis we assume, an arbitrary set of overlapping V^0-to-I^0 chains can occur in these constructions.

13.7.2 A-movement

As we saw in section 13.1, A-movement in English is limited to subject-to-subject or object-to-subject movement. Thus, the landing site is always [Spec,IP]. Further, by minimality of antecedent-government, no

A-link can cross [Spec,IP] without landing there. Thus, whenever two A-chains occur within the same IP, they must both land in the specifier of that IP, that is, they must necessarily intersect. It follows that two well-formed A-chains cannot occur in the same IP; the maximality condition in our definition of chains makes exactly the right judgment and it makes it for exactly the right reason.

13.7.3 Leftward, Non-referential Abar-movement

The analysis here parallels that of A-movement precisely. The landing site for this class of movement, in English, is always [Spec,CP]. Further, $\overline{\text{A}}$-$\overline{\text{Ref}}$-links cannot cross [Spec,CP], again, by minimality of antecedent-government. Two well-formed chains of this class cannot occur, then, in the same CP, and our definition of chains again gets both the judgment and its explanation correct.

13.7.4 Leftward, Referential Abar-movement

Here we can appeal to Manzini's (Manzini 1992) account of $\overline{\text{A}}$-movement which implies that no more than two $\overline{\text{A}}$-chains—one referential and one non-referential—may ever overlap. Consequently, overlapping referential $\overline{\text{A}}$-chains cannot occur. This claim can be argued directly from the apparent distribution of this type of movement in English as well. Consider the following examples (due to Bob Frank):

(16) a. *[$_{\text{NP}_i}$Which car] do you wonder [$_{\text{NP}_j}$which mechanic] John knew [$_{\text{AP}_k}$when] to tell t_j how to fix t_i t_k?
b. ?[$_{\text{NP}_i}$Which car] do you wonder whether John knew [$_{\text{AP}_j}$when] to tell the mechanic how to fix t_i t_j?
c. ?*[$_{\text{NP}_i}$Which car] do you wonder [$_{\text{NP}_j}$which mechanic] John should tell t_j how to fix t_i t_k?
d. ?*[$_{\text{NP}_i}$Which car] do you wonder [$_{\text{NP}_j}$which mechanic] John thought I should tell t_j how to fix t_i t_k?

(17) a. *[$_{\text{NP}_i}$Which painting] do you wonder [$_{\text{NP}_j}$which museum] John decided whether to give t_j t_i?
b. [$_{\text{NP}_i}$Which painting] do you wonder [$_{\text{AP}_j}$how] John decided whether to give t_i the museum t_j?

As is typical of this class of movement, all of these examples are at least somewhat degraded due, presumably, to subjacency violations. The point here is that the cases of overlapping $\overline{\text{A}}$-Ref-movement (Examples 16a, 16c, 16d, 17a) are all at least highly marked, while the corresponding examples in which the overlapping movement involves only one referential chain are relatively acceptable.[10] On this evidence, it would seem that overlapping $\overline{\mathrm{A}}$-Ref-movement results in ill-formed English sentences. Clearly, then, two such chains can never occur in the same domain, and the issue of intersection cannot arise.

13.7.5 Rightward movement

It is in the case of rightward movement that we need to refine our classification. As we saw in our taxonomy of movement, this comes down to five cases (the figure helps in identifying these):

- A complement of a subject raises to adjoin at IP.
- A complement of VP raises to adjoin at VP.
- A complement of a verbal complement raises to adjoin at VP.
- An adjunct of a subject raises to adjoin at IP.
- An adjunct of a verbal complement raises to adjoin at IP.

There is some controversy about the particular site of adjunction in some of these cases, but, as this movement cannot cross CP, the possible number of such sites is bounded, and our approach can be extended to account for any bounded set of sites.

We assume that the number of complements of any phrase is bounded. Thus there is a finite bound on the number of base positions that can participate in the first three cases of rightward movement. We assume further that categories in these positions are distinguished by some new set of features (Comp-1-of-Subject, for instance). Chains formed by these classes of movement, then, will be distinct.

The last two cases present more of a problem. While the base positions involved are all adjoined either to [Spec,IP] or [Comp,VP], either at the XP or $\overline{\mathrm{X}}$ level, there is no bound on the number of categories that can be adjoined in this way. As we saw, though, in the discussion about our treatment of head movement as adjunction (Section 10.3), there is little structural distinction between the nodes adjoined to the same side of the same category. While there may be linguistic reasons for preferring one adjunction structure over another, as long as the theory is based only on structural relationships there will be little to distinguish alternative structures within it. Consider the two structures in Figure 16. With the exception of the inclusion/exclusion relationships between the adjuncts themselves, the structural relationships determined by c-command and domination in the two alternatives are the same. Further, for the most part these adjuncts can be freely permuted. Thus, there is little to prefer one order at D-structure over another (unless one is going to propose

[10] In these examples (31b and 32b) each wh-adverbial is taken to be modifying that clause the [Spec,CP] of which it fills.

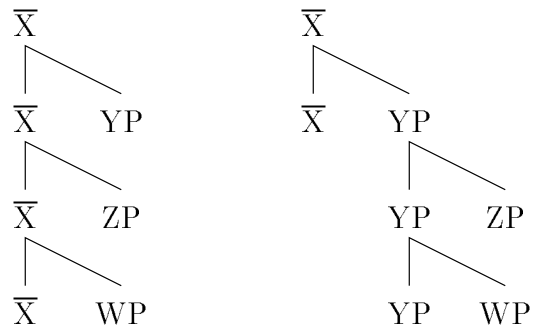

FIGURE 16 Alternate Adjunction Structures

that the adjuncts move among themselves between D-structure and S-structure). We will assume, then, that:

- All right adjunctions to a category are higher than any left adjunction to the same category.
- All adjoined categories that raise from a given position are adjoined at D-structure in the second configuration of Figure 16.
- These all raise simultaneously, in a single movement of the most inclusive category.

This can be refined to allow for movement to multiple target sites from a given D-structure adjunction site. The point is that under these assumptions there are boundedly many pairs of base and target positions, and thus, boundedly many chains suffice. Again, we assume these are distinguished by a finite set of new features.

Note that we end up accepting trees in which the adjunction structure differs superficially from the analysis one is likely to assume, but that are essentially identical structurally to whatever particular analysis that may be. Of course, these structures are all acceptable under X-bar theory, and so we do not need to do anything to explicitly license or require them. The thrust of our treatment is that among the trees we already license, and those that are licensed by standard GB accounts, there is an analysis in which we can account for rightward movement of unboundedly many adjuncts with a bounded number of chains. Our treatment will accept these, and reject others.

13.8 An Example

Figure 17 is the example of Figure 13 with the base, trace, and target categories labeled as we have described. There are two chains: an $\overline{\text{A}}$-referential chain (the NP chain) and an X^0-chain (the I^0-chain). The $\overline{\text{A}}$-

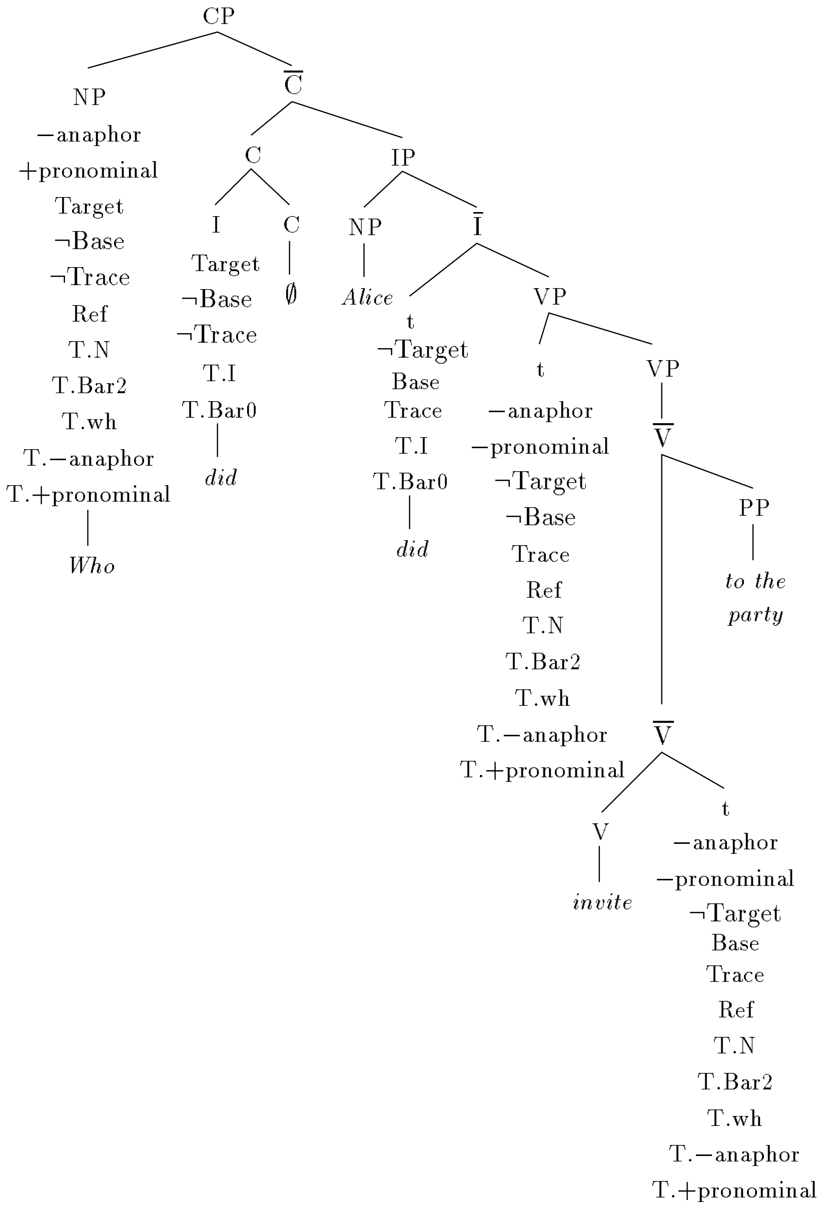

FIGURE 17 Labeled Chains

referential chain is a *wh*-chain. Note that, while its target is pronominal ([−anaphor,+pronominal]) and this determines the value of the corresponding T-features for the chain, the traces do not inherit these features and are [−anaphor,−pronominal] as required by the $\overline{\text{A}}$-Ref-Link relation.

13.9 Discussion

We have sketched, in this chapter, the formalization in $L^2_{K,P}$ of a reasonably complete account of simple movement in English. Since the theory we capture is not parameterized for English but the formalization must necessarily fail for non-context-free natural languages, the question arises of what specializes it to English-like languages. This can be traced to the maximality condition we include in the definition of chains. This condition makes it impossible to identify any of the chains in a configuration in which multiple chains overlap that are not distinguished by class of movement or the features of the target. Thus, the formalization works only if such configurations do not occur. In essence, we must posit a principle ruling out overlapping chains of this sort in order to establish correctness of the formalization. While we argue that such a principle holds for the fragment of English we discuss, we do not presume that it will hold for the class of all natural languages. In fact, we will return to this issue in Chapter 15, where it will become the basis for distinguishing the class of languages for which our formalization succeeds, and which are therefore strongly context-free, from those for which it fails, which are possibly non-context-free.

The fact that this fragment of English can be captured in $L^2_{K,P}$ is a consequence, to a large extent, of the fact that the account we capture, Rizzi's *Relativized Minimality*, is expressed largely in terms of link relations—based on antecedent-government—which have bounded domains. Again, it seems significant that Rizzi has been led by purely linguistic considerations to an account of movement that is nearly exactly tailored for formalization in $L^2_{K,P}$.

Moving from the use of indices to the use of antecedent-government to identify members of chains puts a much greater burden on the definition of antecedent-government, and consequently, on the definition of the link relation. Of course, Rizzi argues that the more complicated notion of antecedent-government is necessary to account for the fact that the different types of antecedent-government do not interact for the purposes of minimality. At the same time, this increase in the complexity of antecedent-government is accompanied by a simplification of the ECP. For Rizzi, the traditional formulation of ECP evolves into a general requirement that every operator be identified with its variable. We push this slightly further, and capture each of Rizzi's cases of identification as link relations. Consequently, for us the traditional formulation of ECP becomes a simple requirement that every category is a member of a (possibly trivial) well-formed chain. In this way, limiting ourselves to a radically impoverished mechanism (CFGs) yields what is in some respects a radically simplified account of a complex array of syntactic effects.

We should emphasize that the formalization we have provided so far can not successfully account for all movement in English. In particular, it is unable to account for structures that Rizzi analyzes under *reconstruction*. The extension of the formalization to account for these cases is the topic of the next chapter.

14

Reconstruction

Our treatment of movement so far works well only as long as no chain is disturbed by subsequent movement. Unfortunately, this is frequently not the case; interactions between movements frequently disrupt the government relations that license their chains. A common example is the case of V-to-I-to-C movement (Figure 14 of the previous chapter). Here, while *has* does c-command its trace, it does not $\overline{\mathrm{X}}$-Antecedent-Govern it. The Bar0 level trace t_j intervenes. This is the worst case of interacting head movement in English and we could make specific exceptions for it, but the same process can occur more generally in other languages. It makes sense to seek a more general solution.

Another case is demonstrated by the sentence

(1) How likely is Alice to win?

We assume the D-structure:[1]

(2) $[_{\mathrm{CP}}\ [_{\mathrm{IP}}$ e is $[_{\mathrm{AP}}\ [_{\mathrm{AP}}$ how] likely $[_{\mathrm{IP}}$ Alice to win]]]]

This transforms by a sequence of four movements:

- *Alice* raises to [Spec,IP] (of the upper IP).
- The subordinate IP raises to adjoin at IP.
- AP raises to [Spec,CP].
- I^0 raises to adjoin at C^0.

The D- and S-structures for this are shown in Figure 18. We are interested in the licensing and identification of the traces, that is, in their proper-head-governors and antecedent-governors (or, rather, their antecedent in the Link relation). In this structure the moved I^0 antecedent-governs its trace, all that is required by the theory. Similarly, the trace of I^0 proper-head-governs the trace of AP, and that trace is antecedent-

[1]This is just one possible analysis. What is important, here, are the interactions of the movements. The details of the analysis are irrelevant.

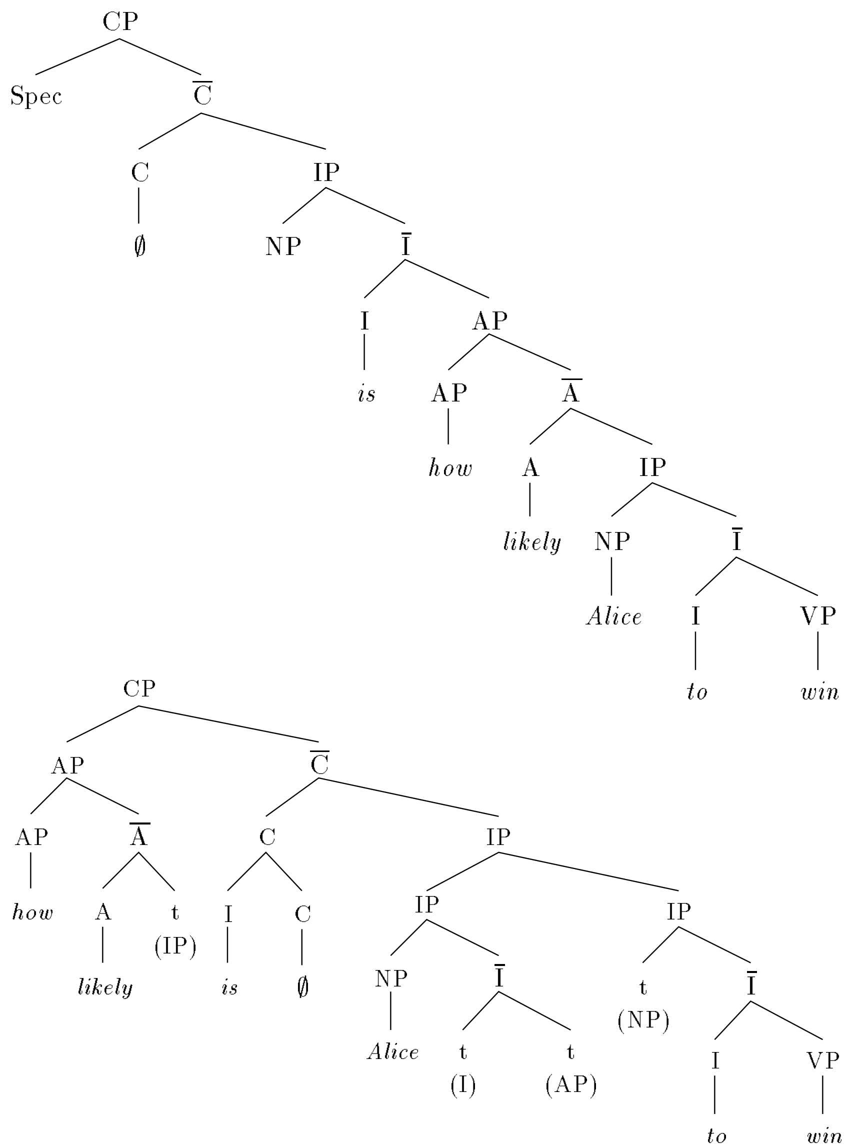

FIGURE 18 Movement of the Base Position

governed by the target position. Also, *likely* proper-head-governs the trace of IP. But that is as far as we get. The moved IP does not even c-command its trace, let alone antecedent-govern it. The same is true of *Alice*. Further, there is no proper-head-governor of the trace of *Alice*.

These are the kinds of difficulties for representational interpretations of movement that are typically handled by reconstruction. Relationships that hold at D-structure but which may be disturbed at S-structure are allowed to hold under reconstruction—in a D-structure that has been reconstructed from the S-structure. This is Rizzi's approach as well.[2] The actual mechanism involved is not often specified, although it is often discussed as if it were a derivational mechanism actually extracting the D-structure embedded in the S-structure.

It is tempting to treat this representationally by interpreting chains, for government relations at least, as single categories. Then the chain governs everything any member of the chain governs. This may well be too strong, and it may be necessary to treat only the Target and Base positions this way. (This is essentially Fong's approach (Fong 1991); the Target position participates in S-structure relations, the Base in D-structure relations.) Unfortunately, this doesn't work for us. We need to identify chains in order to pick out Target/Base pairs, but we need the government relations to identify the chains. Thus, the definition would be circular (or, more precisely, not explicit). The approach is theoretically inviting, though, and can likely be made to work in $L^2_{K,P}$ extended with inductive definitions. In order to stay within the power of $L^2_{K,P}$, we will adopt a more direct, if much less elegant, solution, although one that is still purely declarative, and thus, in keeping with the spirit of representational interpretation.

We distinguish three cases: those in which the Target position is moved, those in which the Base is moved, and those that move an intermediate trace. The moved Target case is the simplest, and, while we could handle it with a mechanism like the one we propose for the other cases, it is possible to treat it more cleanly. Thus, we will deal with this case separately first. We treat the case of moved Bases next, and then

[2]Although he places great importance on a requirement that proper-head-government cannot apply under reconstruction, arguing from evidence in Italian and German. He does not, on the other hand, consider (or license) right movement in English. It is hard to see how the D-structure given for the example can derive the S-structure without either violating the Licensing clause or allowing proper-head-government to apply under reconstruction. We will define the mechanism to handle it. Our definition can easily be restricted to appropriate cases, whatever they may be.

turn to movement of intermediate traces. As we will see, this last case essentially falls under the category of moved Bases.

In all of these situations, what has been disturbed is the locality of the government relation. Our idea is that the notion of locality for each of the government relations will need to be modified to account for the subsequent movement. These modifications, on the other hand, will depend crucially on the original notions of locality. The following predicates extract just the locality conditions—the structural relationships—from the definitions of government in Sections 10.7 and 13.3. These will be used both in defining the mechanisms that account for reconstruction and in redefining the government relations based on those mechanisms.

$$\begin{aligned}\text{Head-Local}(x, y) \quad \equiv \quad & \neg(\exists z)[\text{InterveningBarrier}(z, x, y)] \wedge \\ & \neg(\exists z)[\text{Bar0}(z) \wedge \text{Base}(z) \wedge \text{Intervenes}(z, x, y)]\end{aligned} \tag{1}$$

$$\begin{aligned}&\text{A-Antecedent-Local}(x, y) \equiv \\ &\quad \neg(\exists z)[\text{Intervening-Barrier}(z, x, y)] \wedge \\ &\quad \neg(\exists z)[\text{Spec}(z) \wedge \text{A-pos}(z) \wedge \text{C-Commands}(z, x) \wedge \\ &\qquad \text{Intervenes}(z, x, y)]\end{aligned} \tag{2}$$

$$\begin{aligned}&\overline{\text{A}}\text{-Antecedent-Local}(x, y) \equiv \\ &\quad \neg(\exists z)[\text{Intervening-Barrier}(z, x, y)] \wedge \\ &\quad \neg(\exists z)[\text{Spec}(z) \wedge \neg\text{A-pos}(z) \wedge \text{C-Commands}(z, x) \wedge \\ &\qquad \text{Intervenes}(z, x, y)]\end{aligned} \tag{3}$$

$$\begin{aligned}&\text{Right-Local}(x, y) \equiv \\ &\quad \neg(\exists z, z')[\text{Intervening-Barrier}(z, x, z') \wedge \\ &\qquad\qquad \text{Intervening-Barrier}(z', z, y)]\end{aligned} \tag{4}$$

$$\begin{aligned}&\text{X}^0\text{-Antecedent-Local}(x, y) \equiv \\ &\quad \neg(\exists z)[\text{Intervening-Barrier}(z, x, y)] \wedge \\ &\quad \neg(\exists z)[\text{Bar0}(z) \wedge \text{C-Commands}(z, y) \wedge \text{Intervenes}(z, x, y)]\end{aligned} \tag{5}$$

We also will assume that antecedents c-command their traces asymmetrically (A-C-Command). This is useful, particularly in the analysis of the next section, as it is a transitive relationship.

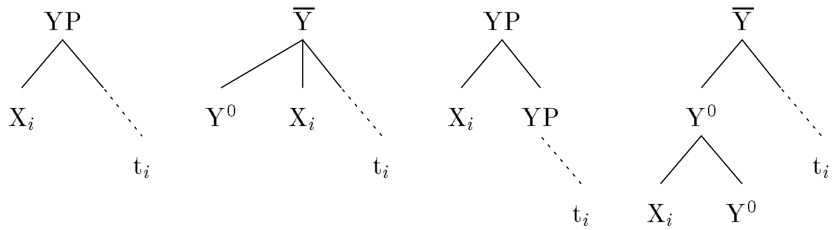

FIGURE 19 Movement of the Target Position

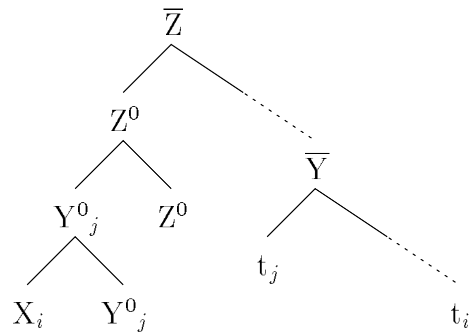

FIGURE 20 Movement of Target—Head Movement

14.1 Moved Targets

Suppose some category X_i (of unspecified Bar level) has moved from a position now occupied by t_i. Then X_i a-c-commands t_i and it is either substituted at a Spec or Comp position of some YP, or adjoined at YP or Y^0.[3] (See Figure 19.) Suppose, further, that some subsequent movement moves X_i. In the first three cases the moved category is either X_i itself, in which case this is a simple case of cyclic movement, or it includes YP (or $\overline{Y}$), and thus, t_i as well.

The only interesting case, then, is the fourth case which occurs when a head with an adjoined head is moved, as in V-to-I-to-C movement in English. Note that any movement that moves X_i without moving Y^0 only forms a new link in the X_i chain. Such movement can be analyzed without reconstruction as simple cyclic movement. Let us suppose, then, that no movement disturbs the relationship between X_i and Y^0. Suppose Y^0 is moved. (The general case, in which any head of a deeper adjunction structure moves, follows by induction.) The result of this movement is shown in Figure 20. Since every head in the adjunction structure includes X_i and no head in that structure dominates it, every category a-

[3] We continue to assume that all head movement is by adjunction.

c-commanded by any head in the structure is a-c-commanded by all such heads. In particular, since Y^0 must a-c-command t_j, then X_i must also a-c-command t_j. Further, t_j must a-c-command t_i (since X_i did when it was attached there). It follows, by the transitivity of a-c-command that X_i still a-c-commands t_i. Thus, movement of the target never disturbs the a-c-command relation of target and trace.

It follows that X_i is local to t_i (that is, it is local when Y^0 is moved back to the position of t_j under reconstruction) iff t_j is local to t_i. Thus, we never need to follow more than one link of the subsequent movements—that of the immediately including head. We can modify our definition of X^0-antecedent-government, then, to pick out a head x that is either X^0-antecedent-local to y itself or is adjoined to a head the base position of which is X^0-antecedent-local to y.

$$
\begin{aligned}
&\mathrm{X}^0\text{-Antecedent-Governs}(x, y) \equiv \\
&\quad \mathrm{Bar0}(x) \wedge \text{A-C-Commands}(x, y) \wedge \mathrm{T.Eq}(x, y) \wedge \\
&\quad \neg\mathrm{Base}(x) \wedge \mathrm{Trace}(y) \wedge \\
&\quad (\exists z)[\ \ (z \approx x \vee \\
&\qquad\quad (\exists z')[\ \ \mathrm{Adj}(x) \wedge z' \lhd x \wedge \mathrm{Target}(z') \wedge \mathrm{Bar0}(z') \wedge \\
&\qquad\qquad\quad \text{A-C-Commands}(z', z) \wedge \mathrm{Base}(z) \wedge \mathrm{T.Eq}(z', z)] \\
&\qquad\quad)\quad \wedge \mathrm{X}^0\text{-Antecedent-Local}(z, y)\quad]
\end{aligned}
\tag{6}
$$

This depends on being able to pick out the appropriate Target/Base pair for the head including X_i on the basis of the T-features. For English, this is guaranteed. All we ever see is a V^0-chain and an I^0-chain. For other languages this is not a valid assumption.

14.2 Moved Base Positions

For moved base positions there is no such simple analysis of the problem. We need a general solution, one that allows the propagation of each of the remaining government relations through arbitrarily many chains. Our approach here is quite artificial. It involves annotating traces in base position with an indication of those traces that can be reached from the target position of its chain (reached in the sense that it is dominated by the target position and local to it). Of course, as long as the locality conditions are met, this will be transitive. The traces that can be reached from the target will include both the traces the target actually dominates and those that can be reached from the targets of those traces.

14.3 Phantoms

Our annotation takes the form of **Phantom** traces. These are nodes outside the X-Bar scheme that are marked as phantoms and occur as children of traces. (Our definition of the X-Bar scheme must be modified to license these.) These are identical in their T-features to the traces they reference. A given trace dominates a phantom for every trace (both ordinary and phantom) that is local to its antecedent. Since there are five notions of locality we need to deal with here (counting both referential and non-referential $\overline{\mathrm{A}}$-movement, but not counting X^0-movement) we will have five varieties of phantoms. We license these in two steps, first defining where they must occur and then requiring them to occur only there.

H-Phantom (7)

$$\begin{array}{ll}(\forall x, x', y')(\exists y)[& (\mathrm{Trace}(x) \wedge \mathrm{Link}(x', x) \wedge x' \triangleleft^{+} y' \wedge \\ & \quad (\mathrm{Target}(x') \wedge \text{Head-Local}(x', y') \wedge \\ & \quad\quad (\mathrm{Trace}(y') \vee \text{H-Phantom}(y')\ ) \vee \\ & \quad \mathrm{Trace}(x')) \wedge \text{H-Phantom}(y') \quad) \rightarrow \\ & x \triangleleft y \wedge \text{H-Phantom}(y) \wedge \mathrm{T.Eq}(y', y) \quad]\end{array}$$

This says that whenever x is a trace with antecedent x' (as identified by Link) and either:

- x' is Target and is Head-Local to a trace y' or
- x' is Target and is Head-Local to an H-Phantom y' or
- x' is a trace and dominates an H-Phantom y',

then there is an H-Phantom y under x that takes its T-features from y'.

The remaining four cases differ only in the variety of the phantom and notion of locality.

A-Phantom (8)

$$\begin{array}{ll}(\forall x, x', y')(\exists y)[& (\mathrm{Trace}(x) \wedge \mathrm{Link}(x', x) \wedge x' \triangleleft^{+} y' \wedge \\ & \quad (\mathrm{Target}(x') \wedge \text{A-Antecedent-Local}(x', y') \wedge \\ & \quad\quad (\mathrm{Trace}(y') \vee \text{A-Phantom}(y')\ ) \vee \\ & \quad \mathrm{Trace}(x')) \wedge \text{A-Phantom}(y') \quad) \rightarrow \\ & x \triangleleft y \wedge \text{A-Phantom}(y) \wedge \mathrm{T.Eq}(y', y) \quad]\end{array}$$

$\overline{\text{A}}$-$\overline{\text{Ref}}$-Phantom (9)

$$
\begin{aligned}
(\forall x, x', y')(\exists y)[\quad & (\text{Trace}(x) \wedge \text{Link}(x', x) \wedge x' \triangleleft^{+} y' \wedge \\
& \quad (\text{Target}(x') \wedge \overline{\text{A}}\text{-Antecedent-Local}(x', y') \wedge \\
& \quad\quad (\text{Trace}(y') \vee \overline{\text{A}}\text{-}\overline{\text{Ref}}\text{-Phantom}(y')\ ) \vee \\
& \quad \text{Trace}(x')) \wedge \overline{\text{A}}\text{-}\overline{\text{Ref}}\text{-Phantom}(y') \quad) \rightarrow \\
& x \triangleleft y \wedge \overline{\text{A}}\text{-}\overline{\text{Ref}}\text{-Phantom}(y) \wedge \text{T.Eq}(y', y) \quad]
\end{aligned}
$$

$\overline{\text{A}}$-Ref-Phantom (10)

$$
\begin{aligned}
(\forall x, x', y')(\exists y)[\quad & (\text{Trace}(x) \wedge \text{Link}(x', x) \wedge x' \triangleleft^{+} y' \wedge \\
& \quad (\text{Trace}(y') \vee \overline{\text{A}}\text{-Ref-Phantom}(y')) \quad) \rightarrow \\
& x \triangleleft y \wedge \overline{\text{A}}\text{-Ref-Phantom}(y) \wedge \text{T.Eq}(y', y) \quad]
\end{aligned}
$$

R-Phantom (11)

$$
\begin{aligned}
& (\forall x, x', y')(\exists y)[\\
& \quad (\text{Trace}(x) \wedge \text{Link}(x', x) \wedge x' \triangleleft^{+} y' \wedge \\
& \quad\quad (\text{Target}(x') \wedge \text{Right-Antecedent-Local}(x', y') \wedge \\
& \quad\quad\quad (\text{Trace}(y') \vee \text{R-Phantom}(y')\ ) \vee \\
& \quad\quad \text{Trace}(x')) \wedge \text{R-Phantom}(y') \quad) \rightarrow \\
& \quad x \triangleleft y \wedge \text{R-Phantom}(y) \wedge \text{T.Eq}(y', y) \quad]
\end{aligned}
$$

The opposite direction is similar:

$$
\begin{aligned}
& (\forall y)[\text{H-Phantom}(y) \rightarrow \\
& \quad \neg(\exists z)[y \triangleleft z] \wedge \\
& \quad \neg(\text{Base}(y) \vee \text{Bar0}(y) \vee \text{Bar1}(y) \vee \text{Bar2}(y) \vee \text{Lexical}(y)) \quad \wedge \\
& \quad (\exists x, x', y')[\\
& \quad\quad x \triangleleft y \wedge \text{T.Eq}(y', y) \wedge \\
& \quad\quad \text{Trace}(x) \wedge \text{Link}(x', x) \wedge x' \triangleleft^{+} y' \wedge \\
& \quad\quad (\text{Target}(x') \wedge \text{Head-Local}(x', y') \vee \text{Trace}(x')) \wedge \\
& \quad\quad (\text{Trace}(y') \vee \text{H-Phantom}(y')) \quad] \quad]
\end{aligned}
\tag{12}
$$

This says that every H-Phantom y is structurally empty (dominates only itself), is not in Base position and is outside the Lexicon and Bar levels, and is the child of a trace x with an antecedent x' that either is a target that is head-local to a trace or H-phantom y' or is a trace dominating an H-Phantom y', and that y and y' agree on their T-features.

Again, the remaining cases vary only in the variety of phantom and notion of locality.

$$
\begin{array}{l}
(\forall y)[\text{A-Phantom}(y) \rightarrow \\
\quad \neg(\exists z)[y \lhd z] \wedge \\
\quad \neg(\text{Base}(y) \vee \text{Bar0}(y) \vee \text{Bar1}(y) \vee \text{Bar2}(y) \vee \text{Lexical}(y)) \quad \wedge \\
\quad (\exists x, x', y')[\\
\qquad x \lhd y \wedge \text{T.Eq}(y', y) \wedge \\
\qquad \text{Trace}(x) \wedge \text{Link}(x', x) \wedge x' \lhd^{+} y' \wedge \\
\qquad (\text{Target}(x') \wedge \text{A-Antecedent-Local}(x', y') \vee \text{Trace}(x')) \wedge \\
\qquad (\text{Trace}(y') \vee \text{A-Phantom}(y')) \quad] \quad]
\end{array}
\tag{13}
$$

$$
\begin{array}{l}
(\forall y)[\overline{\text{A}}\text{-}\overline{\text{Ref}}\text{-Phantom}(y) \rightarrow \\
\quad \neg(\exists z)[y \lhd z] \wedge \\
\quad \neg(\text{Base}(y) \vee \text{Bar0}(y) \vee \text{Bar1}(y) \vee \text{Bar2}(y) \vee \text{Lexical}(y)) \quad \wedge \\
\quad (\exists x, x', y')[\\
\qquad x \lhd y \wedge \text{T.Eq}(y', y) \wedge \\
\qquad \text{Trace}(x) \wedge \text{Link}(x', x) \wedge x' \lhd^{+} y' \wedge \\
\qquad (\text{Target}(x') \wedge \overline{\text{A}}\text{-Antecedent-Local}(x', y') \vee \text{Trace}(x')) \wedge \\
\qquad (\text{Trace}(y') \vee \overline{\text{A}}\text{-}\overline{\text{Ref}}\text{-Phantom}(y')) \quad] \quad]
\end{array}
\tag{14}
$$

$$
\begin{array}{l}
(\forall y)[\overline{\text{A}}\text{-Ref-Phantom}(y) \rightarrow \\
\quad \neg(\exists z)[y \lhd z] \wedge \\
\quad \neg(\text{Base}(y) \vee \text{Bar0}(y) \vee \text{Bar1}(y) \vee \text{Bar2}(y) \vee \text{Lexical}(y)) \quad \wedge \\
\quad (\exists x, x', y')[\\
\qquad x \lhd y \wedge \text{T.Eq}(y', y) \wedge \\
\qquad \text{Trace}(x) \wedge \text{Link}(x', x) \wedge x' \lhd^{+} y' \wedge \\
\qquad\qquad (\text{Trace}(y') \vee \overline{\text{A}}\text{-Ref-Phantom}(y')) \quad] \quad]
\end{array}
\tag{15}
$$

$$
\begin{array}{l}
(\forall y)[\text{R-Phantom}(y) \rightarrow \\
\quad \neg(\exists z)[y \lhd z] \wedge \\
\quad \neg(\text{Base}(y) \vee \text{Bar0}(y) \vee \text{Bar1}(y) \vee \text{Bar2}(y) \vee \text{Lexical}(y)) \quad \wedge \\
\quad (\exists x, x', y')[\\
\qquad x \lhd y \wedge \text{T.Eq}(y', y) \wedge \\
\qquad \text{Trace}(x) \wedge \text{Link}(x', x) \wedge x' \lhd^{+} y' \wedge \\
\qquad (\text{Target}(x') \wedge \text{Right-Antecedent-Local}(x', y') \vee \text{Trace}(x')) \wedge \\
\qquad (\text{Trace}(y') \vee \text{R-Phantom}(y')) \quad] \quad]
\end{array}
\tag{16}
$$

These phantoms provide the means for extending the definitions of antecedent-government. Government holds between x and y if either y or a phantom of y are appropriately local to x.

$$\begin{aligned}&\text{Head-Governs}(x,y) \equiv \\ &\quad (\exists y')[(y \approx y' \vee \text{H-Phantom}(y') \wedge \text{T.Eq}(y,y')) \wedge \\ &\quad \text{Bar0}(x) \wedge \text{M-Command}(x,y') \wedge \\ &\quad (\text{Lexical}(x) \vee \text{T}(x) \vee +\text{agr}(x) \wedge \text{Agreement}(x,y)) \wedge \\ &\quad \text{Head-Local}(x,y')]\end{aligned} \tag{17}$$

$$\begin{aligned}&\text{A-Antecedent-Governs}(x,y) \equiv \\ &\quad (\exists y')[(y \approx y' \vee \text{A-Phantom}(y') \wedge \text{T.Eq}(y,y')) \wedge \\ &\quad \text{A-pos}(x) \wedge \text{C-Commands}(x,y') \wedge \text{T.Eq}(x,y') \wedge \\ &\quad \text{A-Antecedent-Local}(x,y')]\end{aligned} \tag{18}$$

$$\begin{aligned}&\overline{\text{A}}\text{-Antecedent-Governs}(x,y) \equiv \\ &\quad (\exists y')[(y \approx y' \vee \overline{\text{A}}\text{-}\overline{\text{Ref}}\text{-Phantom}(y') \wedge \text{T.Eq}(y,y')) \wedge \\ &\quad \neg\text{A-pos}(x) \wedge \text{C-Commands}(x,y') \wedge \text{T.Eq}(x,y') \wedge \\ &\quad \overline{\text{A}}\text{-Antecedent-Local}(x,y')]\end{aligned} \tag{19}$$

$$\begin{aligned}&\text{Right-Antecedent-Governs}(x,y) \equiv \\ &\quad (\exists y')[(y \approx y' \vee \text{R-Phantom}(y') \wedge \text{T.Eq}(y,y')) \wedge \\ &\quad \text{Adj}(x) \wedge \text{C-Commands}(x,y') \wedge \text{T.Eq}(x,y') \wedge \\ &\quad \text{Right-Antecedent-Local}(x,y')]\end{aligned} \tag{20}$$

This definition of phantoms is in terms of the Link relation, which, in turn, depends on the definition of antecedent-government. This, of course depends on the the distribution of phantoms. So this, then, is an implicit definition, which we can interpret inductively—as more phantoms are licensed, more links will be defined, which in turn licenses more phantoms. Note that the non-monadic predicates involved are still explicitly defined. They resolve to formulae in which Phantom(x) occurs, but in which none of the non-monadic predicates occur. We must establish, then, that it is in fact a well-defined implicit definition of Phantom.

Note that every movement creates one link. Invariably, then, there is at least one link that is undisturbed. This gives the base case for

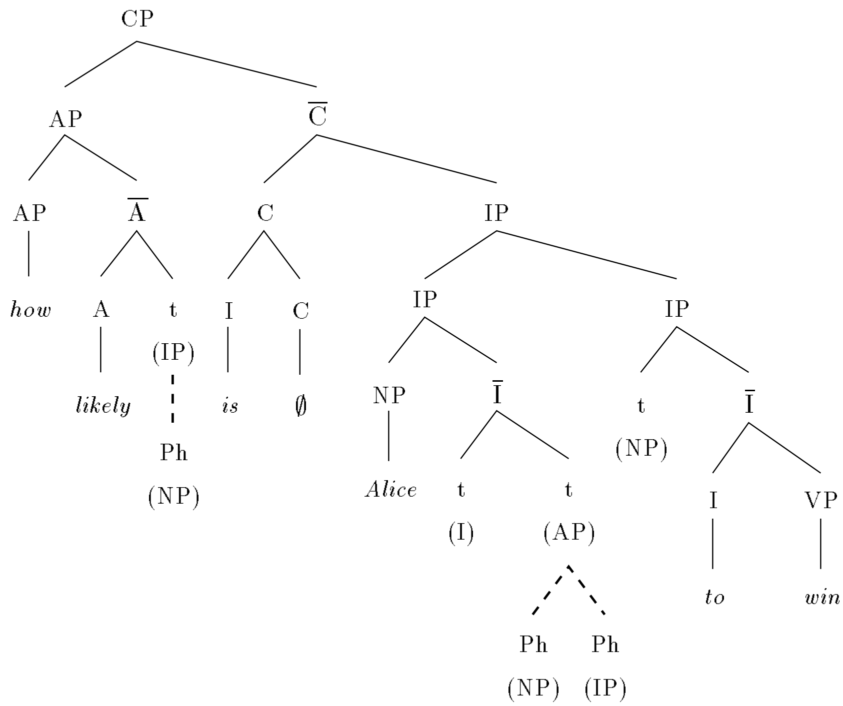

FIGURE 21 Movement with Phantoms

an induction on the number of movements that have affected a chain. For the induction step, note that if all phantoms due to the movements that affect a chain have been properly located, then the link relations for that chain can be determined using at most those phantoms. That the induction terminates follows from the fact that each step correctly sets the phantoms for one movement. No derivable structure involves more than finitely many movements.

As an example of how this works, consider the example shown again in Figure 21, this time decorated with the relevant phantoms. Here the link between the AP and its trace is undisturbed. Since the target of this movement is local to the trace of the IP for both A-antecedent-government, and Right-antecedent-government, there must be one of each of these types of phantoms, each referencing the trace of the IP, attached to the trace of the AP. (For clarity we represent both of these with a single phantom in the figure.) We then have the link between the moved IP and its trace established via this phantom. Since the NP trace is local to the antecedent of this link for both head-government and A-antecedent-government, there must be one of each of these types of phantom, each referencing the trace of the NP, attached to the trace of the IP. Since this is A-antecedent-local to the antecedent of the AP movement, this immediately implies that there is such an A-Phantom attached to the trace of the AP, as well. Note that the AP is not head-local to the H-Phantom at the NP trace, and this phantom does not propagate to the AP trace.

These phantoms are sufficient to resolve the government relations in the figure. The trace of *Alice* is now proper-head-governed by virtue of the fact that *likely* head-governs its phantom under the IP trace. Right-antecedent-government of the IP trace, is established by the IP phantom under the AP trace. And finally, A-antecedent-government of the trace of *Alice* is established via the NP phantom under the AP trace. Note that this last government relation holds through the mediation of two chains—the AP chain and the IP chain—but requires reference to only one phantom. Propagation through any finite number of chains is handled by the definition of Phantom, which can be implicit, rather than the definition of government, which must be explicit.

As with X^0-antecedent-government, this mechanism depends on being able to distinguish pairs related by Link on the basis of their T-features. Unlike that case, unfortunately, there is no simple way of establishing that this in fact can be done. In particular, it is possible, in principle, to move a head out of the local domain of its movement. Thus, we cannot guarantee that chains in otherwise properly formed structures will not interfere. Cases of movement that actually occur in English seem not to suffer from this, but we are not aware of any research that addresses the question of whether they are possible. In the absence of results ruling out such structures, then, we can claim coverage only of those structures in which nested movement does not lead to conflation of chains. Since structures in which such conflation does occur are likely to be at best marginal, this still seems to be an adequate fragment of English.

14.4 Moved Intermediate Traces

We need only account, now, for movement of traces that are neither Target nor Base. This case, for movement of maximal projections, is shown schematically in Figure 22. Note that since the trace itself is invisible to movement, anything that moves it must move YP, and thus, move the Base position as well. This situation, then, is subsumed under the moved Base case.

Moved intermediate traces simply do not occur for head movement,

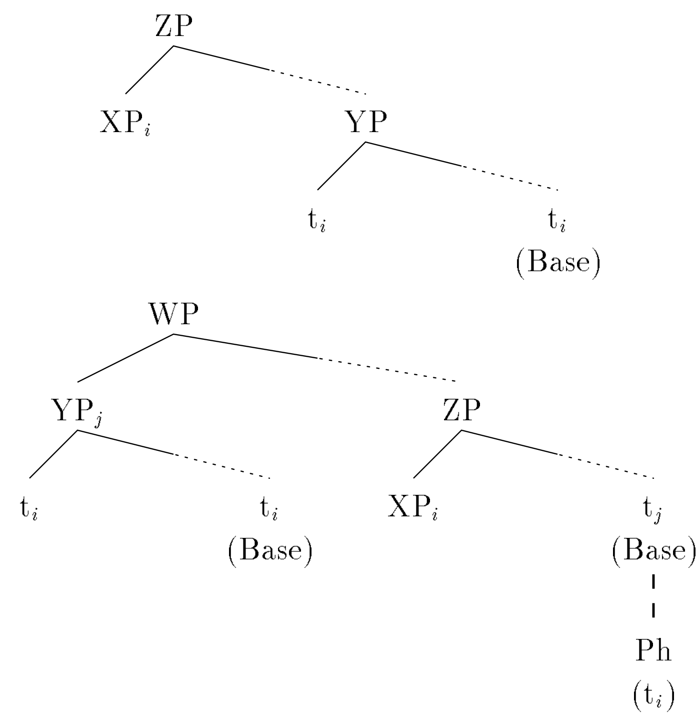

FIGURE 22 Movement of Intermediate XP-Traces

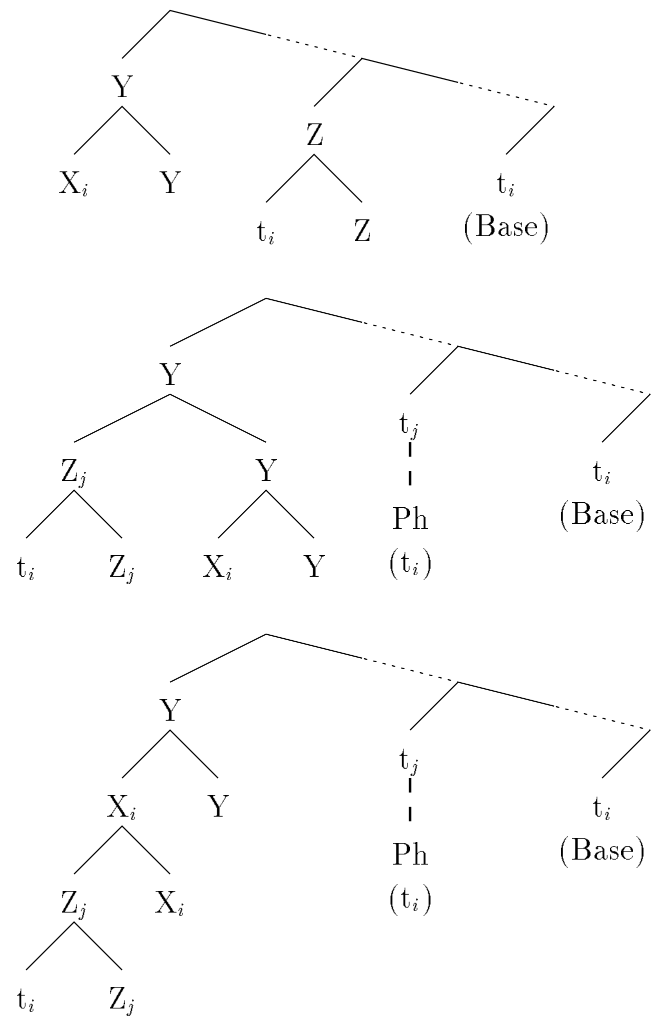

FIGURE 23 Movement of Intermediate X^0-Traces

at least not in English. Nonetheless, Figure 23 shows the appropriate configuration, if such movement could occur. Note that the requirement that all head movement be head-local implies that Y^0 (and thus, X^0) is the closest head c-commanding Z^0, and that movement of Z^0 must adjoin at one of these two. In both these cases, resolution of the government relation between X^0 and the intermediate trace cannot be resolved via the category immediately including X^0. Thus, the X^0-antecedent-government mechanism, as we give it, will not cover these cases. It is simple, on the other hand, to extend the phantom mechanism to include X^0-phantoms as well. Then, as with the case of moved intermediate XP-traces, the moved Base mechanism will resolve the moved intermediate trace case as well.

15

Limitations of the Interpretation

In this chapter we look at two classes of structures that our mechanism cannot accommodate—the analysis of cross-serial dependencies in Dutch due to Bresnan, et al. (1982), and certain long-distance extraction phenomena in Swedish (Miller 1991). The fact that the mechanism breaks down in essentially the same way for both of these classes of structures suggests that the property responsible for the failure may be characteristic of non-context-free natural languages.

15.1 Cross-Serial Dependencies in Dutch

The first class of structures we examine are those of the analysis of cross-serial dependencies in Dutch that Bresnan, et al., have argued are strongly non-context-free (recast in the current theory). The inability of our interpretation to license these structures is due to a failure of head movement in them to be adequately bounded.

The class of sentences of interest are typified by the (slightly modified) example from Bresnan et al. 1982:

(1)

dat	Jan	Piet	Marie	de kinderen	zag	helpen	helpen
that	Jan	Piet	Marie	the children	saw-past	help-inf	help-inf

zwemmen
swim-inf

that Jan saw Piet help Marie help the children swim

A possible D-structure for this is given in Figure 24. This is transformed through a sequence of head movements: a verb adjoins to its INFL which then adjoins to the next higher verb. The result is the S-structure of Figure 25. This is something like an extension of the V-to-I-to-C movement in English to arbitrarily deep structures. Our analysis of moved targets still holds. Each of the targets in Figure 25 a-c-commands its trace, and the trace of the category immediately enclosing the target is still X^0-antecedent-local to its trace. The problem is that the domain

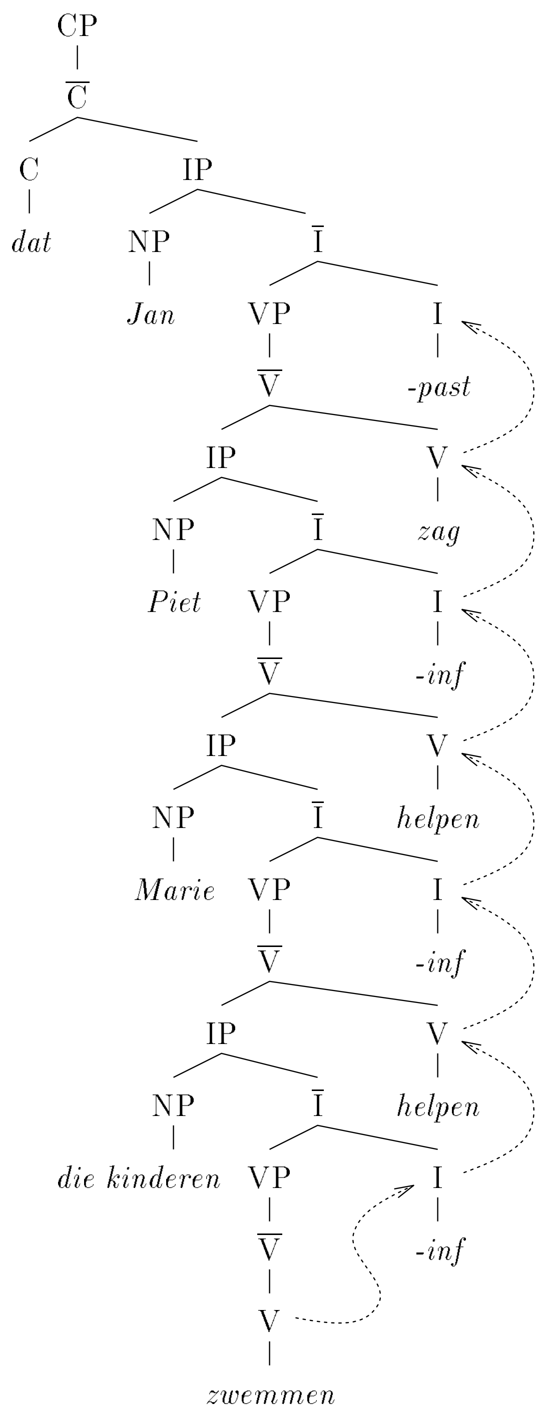

FIGURE 24 Cross-Serial Dependencies—D-Structure

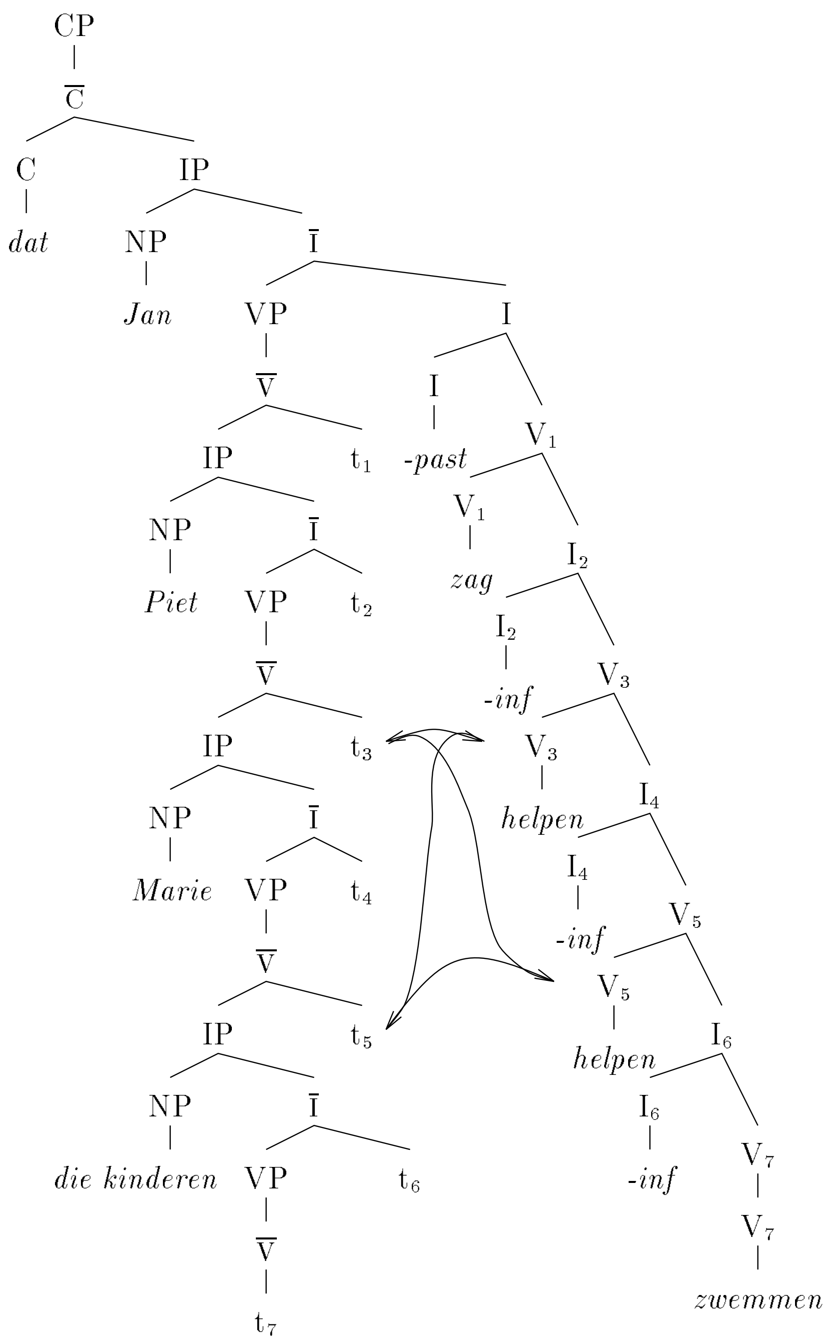

FIGURE 25 Cross-Serial Dependencies—S-Structure

of these movements is no longer bounded and so we can no longer guarantee that the chains can be uniquely identified by their T-features. In fact we can not license either of the *helpen* chains, since the T-features of both instances of *helpen* agree, as do the T-features of their INFLs. Thus, the target of either one X^0-antecedent-governs the base of the other and there is no set that is both linearly ordered and maximal in the way we require chains to be.

15.2 Long-Distance Extraction in Swedish

The second class of structures is less clear cut. These are the long-distance extraction phenomena in Scandinavian languages that Miller argues are non-context-free in Miller 1991.[1]

His example is the sentence (shown in Figure 26):

(2)	Här	är	pojken$_1$	som	jag	undrar	[$_2$vilken flicka]	Kalle
	This	is	[$_1$the boy]	that	I	wonder	[$_2$which girl]	Kalle

undrar	[$_3$vilka barn]	han$_1$	trodde att	hon$_2$
wonders	[$_3$which children]	he$_1$	thinks that	she$_2$

hadde rekommenderat	t$_3$	til studenterna.
had recommended	t$_3$	to the students.

Strictly speaking, this is an agreement issue rather than a movement issue. There is only one actual movement—the $\overline{\text{A}}$-referential-chain of *vilka barn*—and this is non-problematic. The other co-indexations involve resumptive pronouns. These, in fact, are required for Miller's weak generative capacity argument, since the verb inflections are insufficiently distinct.[2] Some kind of identification mechanism is necessary, though, to mediate the agreement between the pronoun and its referent.

The section of the structure that may be iterated is the CP

$[_{\mathrm{CP}}\ [_{\mathrm{NP}}$ vilk– NP] NP undrar CP]

and the agreement of interest is that between the *wh*-NP in [Spec,CP] and its corresponding pronoun. While these are certainly referential in the intuitive sense of Rizzi's distinction, the pronoun is not in a referential position, as it is a subject, and thus, not directly Theta-marked. But we cannot interpret these as something like $\overline{\text{A-Ref}}$-antecedent-government, because the link crosses an intervening $\overline{\text{A}}$-specifier position.[3] We might extend the use of referential indices to

[1] In fact, he argues that these structures are not only non-context-free, but are outside the generative power of indexed languages as well.

[2] In the case of *pojken* (*the boy*) it would seem that there can be no movement anyway, as it would be from subject (*he thinks...*) to object (*... is the boy*) and resulting chain would have multiple Theta-roles.

[3] A corresponding sentence in English might be

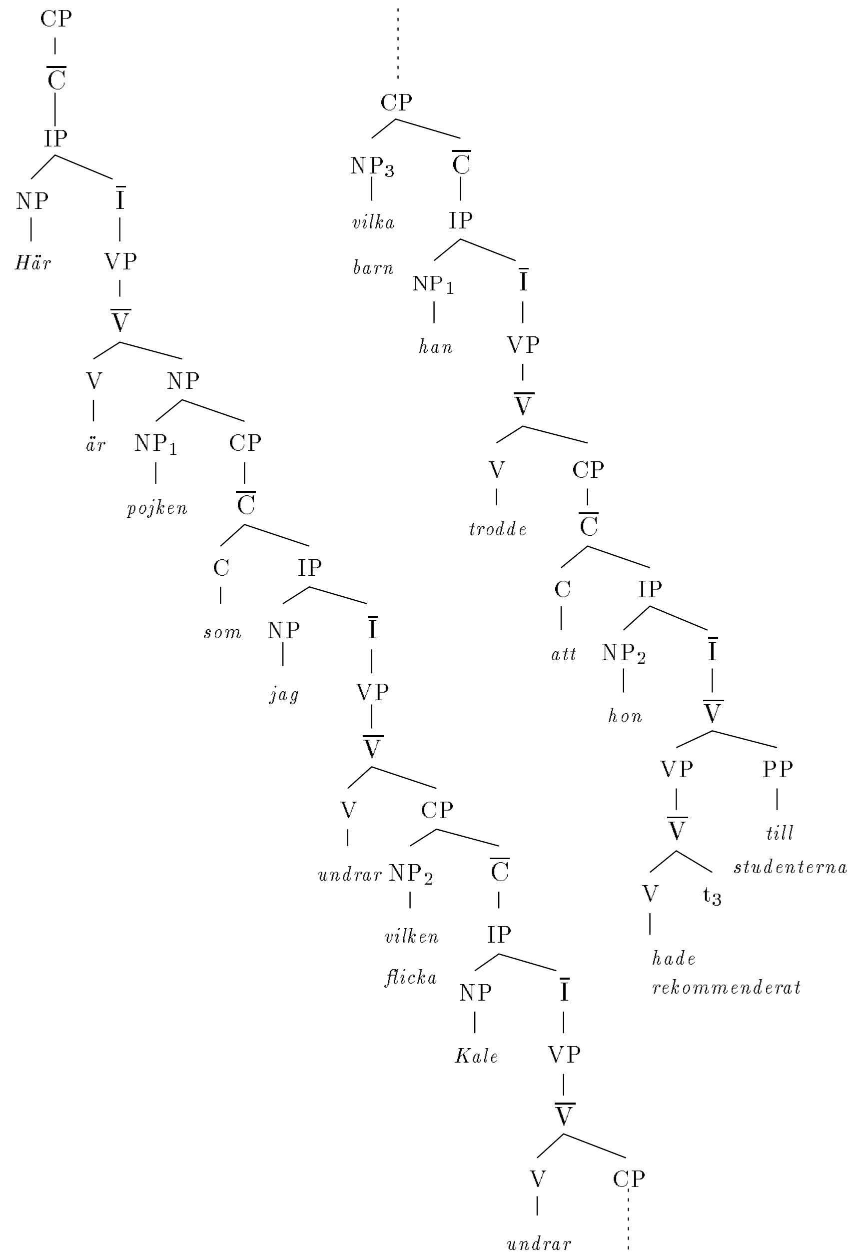

FIGURE 26 Long-Distance Dependencies

accommodate binding resumptive pronouns in a situation like this, but, again, our mechanism fails, in this case because our assumption that referential links never overlap fails. It would appear, then, that there may be no way to handle these structures without being able to distinguish arbitrarily many overlapping chains.

15.3 A Class of "English-like" Languages

In both of these constructions the difficulty for our formalization is the need to distinguish an unbounded number of chains. It is our ability to classify chains in English into a bounded set of types in such a way that no two chains of the same type overlap that is critical to the success of the approach. In fact, our formalization can be adapted to the theory of any language that is consistent with standard GB accounts and in which there is an account of movement that respects such a bound.[4] We can state this as a principle:

> The number of chains which overlap at any single position in the tree is bounded by a constant.

Arguments for the existence of such bounds have appeared in the linguistics literature. As we noted earlier, Manzini's *Locality Theory* (1992) implies that there are never more than two $\overline{\text{A}}$-chains overlapping at any point. Stabler (1994) makes the stronger claim that such bounds exist for all linguistically relevant relationships in all natural languages.

Leaving aside the possibility that it may be possible to account for cross-serial dependencies and long-distance extractions in other ways, it is suggestive that both of these canonical examples of non-context-free constructions in natural language fail to satisfy this principle while, conversely, those languages that do can evidently be formalized, using our approach, in $L^2_{K,P}$ and are consequently strongly context-free. It would seem, then, at least if we restrict attention to languages with potentially finite lexicons (and thus exclude the non-context-free account of Bambara), that within the realm of GB the principle seems to separate the context-free languages from the non-context-free.

> *[Which boy]$_i$ did you wonder [which girl]$_j$ wondered t$_i$ thought t$_j$ recommended the book.

which fails, at least in part, for this reason.

[4] This also provides a perspective on what it means for a relationship to be local. By definition, every relationship we can capture within $L^2_{K,P}$ is local in the sense that it can be enforced by CFGs. A typical interpretation of the notion of local relations is that they involve a bounded domain. But we can capture relations between elements that are unboundedly far apart. What we cannot capture are relationships in which there are unboundedly many overlapping domains. This observation is implicit in Joshi and Levy 1982.

16

Conclusion of Part II

We have explored, in this part of the book, the twin issues of definability and non-definability in $L^2_{K,P}$ of the principles of Government and Binding Theory. When coupled with our result from the first half of the book characterizing the Local sets by definability in $L^2_{K,P}$, these explorations relate directly to the question of which sets of these principles can be enforced by Context-Free Grammars.[1] Thus, we get generative capacity results for formalizations of languages within the GB framework.

We have two main results. The first states that free-indexation, as it is generally interpreted in GB, is not definable in $L^2_{K,P}$. In this interpretation indices are assumed to be assigned randomly, with inappropriately indexed structures being filtered out by some set of constraints on the relationships between co-indexed elements. The immediate implication of the non-definability in $L^2_{K,P}$ of this form of indexation is that it is capable of expressing conditions on the phrase markers that cannot be enforced by CFGs. Our result, though, is considerably stronger than this. We have shown that this approach to indexation is capable of defining sets of trees for which emptiness is undecidable, even when the constraints on the indexing are severely restricted—limited to agreement conditions and constraints stated in terms of pairs of nodes related by one or two levels of immediate domination. This suggests that, while free-indexation is a conceptually simple means of expressing certain structural principles, it is perhaps too powerful for formal theories of language, at least if one hopes to be able to establish the consistency of those theories. This is not an unprecedented idea. We cite Chomsky (1993) questioning the appropriateness of indices as fun-

[1] While this is the typical characterization of the local sets, it should not be taken too literally, especially when considering the consequences for the nature of the human language faculty. As we noted in Chapter 1, in that context the key characteristic of the local sets, perhaps, is that they are accepted by mechanisms that are equivalent to finite-state tree automata.

damental entities in linguistic theories, and suggesting that it should be possible to dispense with them in favor of direct expressions of the structural relationships they capture.

The second of our central results states that a specific set of principles commonly employed in GB accounts of language, when extended slightly, is definable in $L^2_{K,P}$, and that this set of principles embodies a substantially complete GB account of common English syntax. Again, the superficial consequence of this result is a claim that English is a context-free language. It seems more significant, though, that the basis of this claim is the claim that a fairly comprehensive formalization of English within the GB framework is strongly context-free. This is quite a strong language complexity result for a theoretical framework in which such results are extremely difficult to come by. Thus, it serves as an example of the power of our characterization of the Local sets in establishing results of this type.

As we noted in our introductory comments, formalizations of linguistic theories often have much to offer those theories. In developing our formalization we have sketched a few examples of these potential uses. Perhaps the most important benefit is the ability to verify aspects of the theory formally. As a trivial example of how such verification can be carried out we formally derive, within our definition of binding theory, the fact that PRO must be ungoverned from the assumption that it is [+anaphor,+pronominal]. This, of course, is an elementary result in GB, and the exercise probably has as much value in verifying our formalization as it has in verifying the result. Nonetheless, it is an indication of the way in which such predictions can be given a rigorous foundation.

The need for such rigorous foundations is illustrated by a second benefit formalizations such as this one can provide for linguistic theories. Frequently the process of formalization will clarify details of the theory that are incomplete, have been overlooked, or simply cannot be determined in a less precise context. An example is the fact that the *Barriers* definition of c-command allows for, but does not require, mutual and reflexive c-command. While Chomsky raises the possibility of restricting this, there is little reason in that context to choose between the possible interpretations of the relationship. In our work here, on the other hand, we point out that c-command is formally better behaved if it is assumed to be asymmetric. This assumption, in fact, plays a role in our analysis of the effects of subsequent movement on chains. As it turns out, this observation, as with our observations about indexation, has been anticipated in the GB literature. Kayne (1994) employs asymmetric c-command in deriving X-Bar structure from the linear ordering of terminal strings.

This is a minor example of the way in which the needs of the formalization can illuminate issues that have linguistic significance beyond that formalization. A wider array of such issues arise from the fact that we cannot capture indexation directly. This has led us to explore the extent to which the principles of GB actually employ indexation necessarily. One of the things these explorations have highlighted is a distinction between those principles of binding theory and control theory that govern the distribution of nominals (Principle A and Obligatory Control) and those principles that only govern their interpretation (Principles B and C, and Optional Control). For binding theory, at least, this distinction is discernible in the theory developed in Chomsky 1993, as well. More significant, though, is the fact that we have shown that those principles that actually govern distribution of the nominals can be expressed without the use of indexation, and that this can be done in a natural way.

The problem of capturing the chains formed by movement without using indexation is more substantial. Our approach is to look at chains as linear sequences of link relations of a restricted sort. Here again, our concerns are paralleled by issues raised in the GB literature, particularly in Rizzi's *Relativized Minimality* (1990) and Manzini's *Locality Theory* (1992). To a large extent we owe our success in capturing chains to Rizzi's account, which is couched largely in terms of the antecedent-government relation. This raises yet another way in which formalizations can inform the linguistic theory they seek to capture. Rizzi, in developing his account of movement in *Relativized Minimality* is led to a reformulation of the Empty Category Principle (ECP) as two principles—a *licensing* principle that more or less governs the existence of traces, and an *identification* principle that governs their distribution and interpretation. It is this second principle that accounts for most of the extraction asymmetries that are usually attributed to ECP. In our account, we are led by our emphasis on the link relation to take this a step further, and for us the identification principle reduces to a simple requirement that every category is a member of a (possibly trivial) well-formed chain.

Thus, we are led, by the purely internal requirements of our formalization, to a highly simplified account of a wide range of phenomena. In this way, the process of formalizing a theory may suggest alternatives or extensions to the analysis the theory embodies. Another example has to do with the principle that, in Chapter 15, we suggest separates the context-free GB languages from the non-context-free GB languages. If one had an analysis of all natural language structures in which one could bound the number of overlapping chains then one could make a claim for the context-freeness of natural language by proposing the principle as a component of Universal Grammar. The point, here, is not to make such a claim, but to illustrate the way in which the formalization can identify extensions to the theory that have well-defined formal consequences.

Finally, the prospect of being able to identify the formal consequences of extensions to the theory raises the possibility of establishing generative capacity results for (some restriction of) GB as a whole. In a sense this work provides something of a prototype result of this kind. The difficulty of establishing language complexity results for GB can be traced to the fact that the only restrictions on the principles are relatively weak notions, like learnability, coupled with subjective notions like generality, parsimony, and elegance. If these are augmented by formal restrictions of the sort we have developed the result will be a formalism within the realm of GB with non-trivially restricted generative capacity. Our prototype result, then, is that GB, when restricted to sets of principles that are definable in $L^2_{K,P}$, generates only context-free languages. It is only prototypical because we do not actually expect this restriction of GB to be able to generate the entire class of natural languages. Nonetheless, it seems likely that restrictions of the type we employ here, but with somewhat greater generative capacity, could provide non-trivial bounds on the generative capacity of GB theories without compromising their ability to capture the entire range of natural languages.

The question remains of why there should be any correspondence between restrictions on principles of the sort we propose for language complexity reasons and the intuitive notions that drive the development of GB theory. It would be hard to justify restrictions that were wholly artificial from a linguistic point of view. It is here that we believe the significance of the parallels between the issues we have encountered in capturing principles in our restricted formalism and issues that have arisen on purely linguistic grounds in the GB literature lies. If these parallels are not purely coincidental, and we believe they are not, then they suggest that there is a deeper connection between our language complexity concerns and these linguistic intuitions. And so we come full circle. If such a deeper connection exists, it is because the regularities of natural language, and thus the characteristics of the human language faculty, can be distinguished, in part, by the structural properties of language-theoretic complexity classes and their automata-theoretic characterizations.

A

Index of Definitions

Bibliography

Aoun, Joseph, and Robin Clark. 1985. On Non-Overt Operators. In *Southern California Occasional Papers in Linguistics No. 10*. 17–36. University of Southern California.

Bach, Emmon, and William Marsh. 1987. An Elementary Proof of the Peters-Ritchie Theorem. In Savitch et al. Savitch et al. 1987, 41–55.

Backofen, Rolf, James Rogers, and K. Vijay-Shanker. 1995. A First-Order Axiomatization of the Theory of Finite Trees. *Journal of Logic, Language, and Information* 4:5–39.

Berwick, Robert C. 1984. Strong Generative Capacity, Weak Generative Capacity, and Modern Linguistic Theories. *Computational Linguistics* 10:189–202.

Berwick, Robert C. 1991. Principles of Principle-Based Parsing. In Berwick et al. Berwick et al. 1991, 1–37.

Berwick, Robert C., Steven P. Abney, and Carol Tenny (ed.). 1991. *Principle-Based Parsing: Computation and Psycholinguistics*. Kluwer Academic Publishers.

Berwick, Robert C., and Amy S. Weinberg. 1984. *The Grammatical Basis of Linguistic Performance*. MIT Press.

Beth, Evert W. 1953. On Padoa's Method in the Theory of Definition. *Indagationes Mathematicae* 15:330–339.

Blackburn, Patrick, Claire Gardent, and Wilfried Meyer-Viol. 1993. Talking About Trees. In *EACL 93*, 21–29. EACL.

Bresnan, Joan, Ronald M. Kaplan, Stanley Peters, and Annie Zaenen. 1982. Cross-Serial Dependencies in Dutch. *Linguistic Inquiry* 13:613–635.

Brody, Michael. 1993. Θ-Theory and Arguments. *Linguistic Inquiry* 24(1):1–23.

Büchi, J. R. 1960. Weak Second-Order Arithmetic and Finite Automata. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* 6:66–92.

Chandra, Ashok K., and David Harel. 1982. Structure and Complexity of Relational Queries. *Journal of Computer and System Sciences* 25:99–128.

Chomsky, Noam. 1957. *Syntactic Structures*. Janua Linguarum. Mouton and Co. No. IV.

Chomsky, Noam. 1959. On Certain Formal Properties of Grammars. *Information and Control* 2:137–167.

Chomsky, Noam. 1965. *Aspects of the Theory of Syntax*. MIT Press.

Chomsky, Noam. 1982. *Some Concepts and Consequences of the Theory of Government and Binding*. Cambridge, MA: MIT Press.

Chomsky, Noam. 1986. *Barriers*. MIT Press.

Chomsky, Noam. 1993. A Minimalist Program for Linguistic Theory. In *The View from Building 20*. 1–52. MIT Press.

Cornell, Thomas Longacre. 1992. *Description Theory, Licensing Theory, and Principle-Based Grammars and Parsers*. Doctoral dissertation, University of California Los Angeles.

Correa, Nelson. 1991. Empty Categories, Chain Binding, and Parsing. In Berwick et al. Berwick et al. 1991, 83–121.

Courcelle, Bruno. 1983. Fundamental Properties of Infinite Trees. *Theoretical Computer Science* 25:95–169.

Culy, Christopher. 1985. The Complexity of the Vocabulary of Bambara. *Linguistics and Philosophy* 8:345–351.

Doner, John. 1970. Tree Acceptors and Some of Their Applications. *Journal of Computer and System Sciences* 4:406–451.

Elgot, C. C. 1961. Decision Problems of Finite Automata Design and Related Arithmetics. *Transactions of the American Mathematical Society* 98:21–51.

Enderton, Herbert B. 1972. *A Mathematical Introduction to Logic*. Academic Press.

Fong, Sandiway. 1991. *Computational Properties of Principle-Based Grammatical Theories*. Doctoral dissertation, MIT.

Frank, Robert. 1991. Parasitic Gaps and Locality Conditions. In *Papers from the 27th Meeting of the Chicago Linguistic Society*. Chicago Linguistic Society.

Gazdar, Gerald, Ewan Klein, Geoffrey Pullum, and Ivan Sag. 1985. *Generalized Phrase Structure Grammar*. Harvard University Press.

Gorn, Saul. 1967. Explicit Definitions and Linguistic Dominoes. In *Systems and Computer Science, Proceedings of the Conference held at Univ. of Western Ontario, 1965*, ed. John F. Hart and Satoru Takasu. Univ. of Toronto Press.

Gurevich, Yuri. 1984. Toward Logic Tailored for Computational Complexity. In *Computation and Proof Theory*, ed. et al. M. M. Richter. Springer Lecture Notes in Mathematics, Vol. 1104, 175–216. Springer Verlag.

Gurevich, Yuri. 1988. Logic and the Challenge of Computer Science. In *Current Trends in Theoretical Computer Science*, ed. E. Börger. Chap. 1, 1–57. Computer Science Press.

Haegeman, Liliane. 1991. *Government and Binding Theory*. Blackwell.

Hopcroft, John E., and Jeffrey D. Ullman. 1979. *Introduction to Automata Theory, Languages, and Computation*. Reading, Mass.: Addison-Wesley.

Johnson, Mark. 1989. The Use of Knowledge of Language. *Journal of Psycholinguistic Research* 18(1):105–128.

Joshi, Aravind K. 1987. An Introduction to Tree Adjoining Grammars. In *Mathematics of Language*, ed. Alexis Manaster-Ramer. 87–114. John Benjamins.

Joshi, Aravind K., L. S. Levy, and M. Takahashi. 1975. Tree Adjunct Grammars. *Journal of Computer and System Sciences* 10:136–163.

Joshi, Aravind K., and Leon S. Levy. 1977. Constraints on Structural Descriptions: Local Transformations. *SIAM Journal of Computing* 6(2):272–284.

Joshi, Aravind K., and Leon S. Levy. 1982. Phrase Structure Trees Bear More Fruit than You Would Have Thought. *American Journal of Computational Linguistics* 8(1):1–11.

Kayne, Richard S. 1994. *The Antisymmetry of Syntax*. Cambridge, MA: MIT Press.

Koster, Jan. 1987. *Domains and Dynasties*. Dordrecht, Holland: Foris Publications.

Kracht, Marcus. 1995. Syntactic Codes and Grammar Refinement. *Journal of Logic, Language, and Information* 4:41–60.

Kroch, Anthony S., and Aravind K. Joshi. 1987. Analyzing Extraposition in a Tree Adjoining Grammar. In *Syntax and Semantics*. 107–149. Academic Press. Vol. 20.

Läuchli, Hans, and Christian Savioz. 1987. Monadic Second Order Definable Relations on the Binary Tree. *The Journal of Symbolic Logic* 52:219–226.

Lewis, Harry R. 1979. *Unsolvable Classes of Quantificational Formulas*. Addison-Wesley.

Maher, Michael J. 1988. Complete Axiomatizations of the Algebras of Finite, Rational and Infinite Trees. In *Proceedings of the 3rd Annual Symposium on Logic in Computer Science*, 348–457. Edinburgh, Scotland, July.

Manzini, Maria Rita. 1992. *Locality: A Theory and Some of Its Empirical Consequences*. Cambridge, Ma: MIT Press.

McCawley, James D. 1968. Concerning the Base Component of a Transformational Grammar. *Foundations of Language* 4:243–269.

Miller, Philip H. 1991. Scandinavian Extraction Phenomena Revisited: Weak and Strong Generative Capacity. *Linguistics and Philosophy* 14:101–113.

Moschovakis, Yiannis N. 1974. *Elementary Induction on Abstract Structures*. Amsterdam: North-Holland.

Partee, Barbara, Alice ter Meulen, and Robert Wall. 1990. *Mathematical Methods in Linguistics*. Studies in Linguistics and Philosophy, Vol. 20. Dordrecht: Kluwer.

Perrault, C. Raymond. 1984. On the Mathematical Properties of Linguistic Theories. *Computational Linguistics* 10:165–176.

Peters, P. Stanley, and Robert W. Ritchie. 1969. Context-Sensitive Immediate Constituent Analysis—Context-Free Languages Revisited. In *Proceedings of the ACM Conference on the Theory of Computing*, 1–8. ACM, May.

Peters, P. Stanley, and Robert W. Ritchie. 1973. On the Generative Power of Transformational Grammars. *Information Science* 6:49–83.

Peters, Jr., P. Stanley, and R. W. Ritchie. 1971. On Restricting the Base Component of Transformational Grammars. *Information and Control* 18:483–501.

Peters, Jr., P. Stanley, and Robert W. Ritchie. 1973. Context-Sensitive Immediate Constituent Analysis: Context-Free Languages Revisited. *Mathematical Systems Theory* 6(4):324–333.

Pollock, Jean-Yves. 1989. Verb Movement, Universal Grammar and the Structure of IP. *Linguistic Inquiry* 20:365–424.

Pullum, Geoffery K. 1984. On Two Recent Attempts to Show that English Is Not a CFL. *Computational Linguistics* 10:182–188.

Pullum, Geoffrey K., and Gerald Gazdar. 1982. Natural Language and Context-Free Languages. *Linguistics and Philosophy* 4:471–504.

Rabin, Michael O. 1969. Decidability of Second-Order Theories and Automata on Infinite Trees. *Transactions of the American Mathematical Society* 141:1–35.

Rabin, Michael O. 1972. *Automata on Infinite Objects and Church's Problem.* Regional Conference Series in Mathematics, Vol. 13. Providence, RI: American Mathematical Society.

Radford, Andrew. 1988. *Transformational Grammar: A First Course.* Cambridge University Press.

Rizzi, Luigi. 1990. *Relativized Minimality.* MIT Press.

Rogers, James. 1994. *Studies in the Logic of Trees with Applications to Grammar Formalisms.* Ph.D. dissertation, Univ. of Delaware.

Rounds, William C. 1970. Mappings and Grammars on Trees. *Mathematical Systems Theory* 4:257–287.

Savitch, Walter J. 1987. Context-Sensitive Grammar and Natural Language Syntax. In Savitch et al. Savitch et al. 1987, 358–368.

Savitch, Walter J., Emmon Bach, William Marsh, and Gila Safran-Naveh (ed.). 1987. *The Formal Complexity of Natural Language.* D. Reidel Publishing Company.

Shieber, Stuart M. 1985. Evidence Against the Context-Freeness of Natural Language. *Linguistics and Philosophy* 8:333–343.

Siefkes, Dirk. 1978. An Axiom System for the Weak Monadic Second Order Theory of Two Successors. *Israel Journal of Mathematics* 30(3):264–284.

Stabler, Edward P. 1994. The Finite Connectivity of Linguistic Structure. In *Perspectives on Sentence Processing*, ed. C. Clifton, L. Frazier, and K. Rayner. Chap. 13, 303–336. Hillsdale, NJ: Lawrence Erlbaum.

Stabler, Jr., Edward P. 1992. *The Logical Approach to Syntax.* Bradford.

Thatcher, J. W. 1967. Characterizing Derivation Trees of Context-Free Grammars through a Generalization of Finite Automata Theory. *Journal of Computer and System Sciences* 1:317–322.

Thatcher, J. W., and J. B. Wright. 1968. Generalized Finite Automata Theory with an Application to a Decision Problem of Second-Order Logic. *Mathematical Systems Theory* 2(1):57–81.

Thomas, Wolfgang. 1990. Automata on Infinite Objects. In *Handbook of Theoretical Computer Science, Volume B.* Chap. 4, 135–191. Elsevier.

Vijay-Shanker, K. 1987. *A Study of Tree Adjoining Grammars.* Doctoral dissertation, Department of Computer and Information Science, University of Pennsylvania.

Vijay-Shanker, K., David J. Weir, and Aravind K. Joshi. 1987. On the Progression from Context-Free to Tree Adjoining Languages. In *Mathematics of Language*, ed. A. Manaster-Ramer. 389–401. Amsterdam: John Benjamins.

Weir, David J. 1987. Context-Free Grammars to Tree Adjoining Grammars and Beyond. Technical report. University of Pennsylvania.