IRCS Technical Reports Series

Institute for Research in Cognitive Science

May 1995

On Descriptive Complexity, Language Complexity, and GB

James Rogers
University of Pennsylvania

University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS-95-14.

On Descriptive Complexity, Language Complexity, and GB

Abstract
We introduce $L^2_{K,P}$, a monadic second-order language for reasoning about trees which characterizes the strongly Context-Free Languages in the sense that a set of finite trees is definable in $L^2_{K,P}$, if it is (modulo a projection) a Local Set-the set of derivation trees generated by a CFG. This provides a flexible approach to establishing language-theoretic complexity results for formalisms that are based on systems of well-formedness constraints on trees. We demonstrate this technique by sketching two such results for Government and Binding Theory. First, we show that *free-indexation*, the mechanism assumed to mediate a variety of agreement and binding relationships in GB, is not definable in $L^2_{K,P}$, and therefore not enforceable by CFGs. Second, we show how, in spite of this limitation, a reasonably complete GB account of English can be defined in $L^2_{K,P}$. Consequently, the language licensed by that account is strongly context-free. We illustrate some of the issues involved in establishing this result by looking at the definition, $L^2_{K,P}$, of chains. The limitations of this definition provide some insight into the types of natural linguistic principles that correspond to higher levels of language complexity. We close with some speculation on the possible significance of these results for generative linguistics.

Comments
University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS-95-14.

The Institute For Research In Cognitive Science

On Descriptive Complexity, Language Complexity, and GB

by

James Rogers

University of Pennsylvania
3401 Walnut Street, Suite 400C
Philadelphia, PA 19104-6228

May 1995

Site of the NSF Science and Technology Center for Research in Cognitive Science

University of Pennsylvania
Founded by Benjamin Franklin in 1740

IRCS Report 95-14

On Descriptive Complexity, Language Complexity, and GB[1]

James Rogers
Institute for Research in Cognitive Science
University of Pennsylvania
Suite 400C, 3401 Walnut St
Philadelphia, PA 19104-6228, USA
jrogers@linc.cis.upenn.edu

May 24, 1995

[1]To Appear: Specifying Syntactic Structures (papers from the Logic, Structures, and Syntax workshop, Amsterdam, Sept. 1994)

Abstract

We introduce $L^2_{K,P}$, a monadic second-order language for reasoning about trees which characterizes the strongly Context-Free Languages in the sense that a set of finite trees is definable in $L^2_{K,P}$ iff it is (modulo a projection) a Local Set—the set of derivation trees generated by a CFG. This provides a flexible approach to establishing language-theoretic complexity results for formalisms that are based on systems of well-formedness constraints on trees. We demonstrate this technique by sketching two such results for Government and Binding Theory. First, we show that *free-indexation*, the mechanism assumed to mediate a variety of agreement and binding relationships in GB, is not definable in $L^2_{K,P}$ and therefore not enforcible by CFGs. Second, we show how, in spite of this limitation, a reasonably complete GB account of English can be defined in $L^2_{K,P}$. Consequently, the language licensed by that account is strongly context-free. We illustrate some of the issues involved in establishing this result by looking at the definition, in $L^2_{K,P}$, of chains. The limitations of this definition provide some insight into the types of natural linguistic principles that correspond to higher levels of language complexity. We close with some speculation on the possible significance of these results for generative linguistics.

1 Introduction

One of the more significant developments in generative linguistics over the last decade has been the development of *constraint-based* formalisms—grammar formalisms that define languages not in terms of the derivations of the strings in the language, but rather in terms of well-formedness conditions on the structures analyzing their syntax. Because traditional notions of language complexity are generally defined in terms of rewriting mechanisms, complexity of the languages licensed by these formalisms can be difficult to determine.

A particular example, one that will be a focus of this paper, is Government and Binding Theory. While this is often modeled as a specific range of Transformational Grammars, the connection between the underlying grammar mechanism and the language a given GB theory licenses is quite weak. In an extreme view, one can take the underlying mechanism simply to generate the set of all finite trees (labeled with some alphabet of symbols)[1] while the grammatical theory is actually embodied in a set of principles that filter out the ill-formed analyses. As a result, it has been difficult to establish language complexity results for GB theories, even at the level of the recursive [Lap77, Ber84] or context-sensitive [BW84] languages.

That language complexity results for GB should be difficult to come by is hardly surprising. The development of GB coincided with the abandonment, by GB theorists, of the presumption that the traditional language complexity classes would provide any useful characterization of the human languages. This followed, at least in part, from the recognition of the fact that the structural properties that characterize natural languages as a class may well not be those that can be distinguished by existing language complexity classes. There was a realization that the theory needed to be driven by the regularities identifiable in natural languages, rather than those suggested by abstract mechanisms. Berwick characterized this approach as aiming to "discover the properties of natural languages first, and then characterize them formally." [Ber84, pg. 100]

But formal language theory still has much to offer to generative linguistics. Language complexity provides one of the most useful measures with which to compare languages and language formalisms. We have an array of results establishing the boundaries of these classes, and, while many of the results do not seem immediately germane to natural languages, even seemingly artificial diagnostics (like the copy language $\{ww \mid w \in (ab)^*\}$) can provide the basis for useful classification results (such as Shieber's argument for the non-context-freeness of Swiss-German [Shi85]). More importantly, characterization results for language complexity classes tend to be in terms of the *structure* of languages, and the structure of natural language, while hazy, is something that can be studied more or less directly. Thus there is a realistic expectation of finding empirical evidence falsifying a given hypothesis. (Although such evidence may well be difficult to find, as witnessed by the history of less successful attempts to establish results such as Shieber's [PG82, Pul84].) Further, language complexity classes characterize, along one dimension, the *types* of resources necessary to parse or recognize a language. Results of this type for the class of human languages, then, make specific predictions about the nature of the human language faculty, predictions that, at least in principle, can both inform and be informed by progress in uncovering the physical

[1] Or, following a strictly derivational approach, the set of all structures consisting of a triple of finite trees along with a representation of PF.

nature of that faculty.

In this paper we discuss a flexible and quite powerful approach to establishing language complexity results for formalisms based on systems of constraints on trees. In Section 2 we introduce a logical language, $L^2_{K,P}$, capable of encoding such constraints lucidly. The key merit of such an encoding is the fact that sets of trees are definable in $L^2_{K,P}$ if and only if they are strongly context-free. Thus definability in $L^2_{K,P}$ characterizes the strongly context-free languages. This is our primary result, and we develop it in Section 3.

We have used this technique to establish both inclusion and exclusion results for a variety of linguistic principles within the GB framework [Rog94]. In the remainder of the paper we demonstrate some of these. In Section 4 we sketch a proof of the non-definability of free-indexation, a mechanism that is nearly ubiquitous in GB theories. The consequence of this result is that languages that are licensed by theories that necessarily employ free-indexation are outside of the class of CFLs. Despite the unavailability of free-indexation, we are able to capture a mostly standard GB account of English within $L^2_{K,P}$. Thus we are able to show that the language licensed by this particular GB theory is strongly context-free. In Section 5 we illustrate some of the issues involved in establishing this result, particularly in light of the non-definability of free-indexation. We close, finally, with some speculation on the possible significance of these results for generative linguistics.

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

The idea of employing mathematical logic to provide a precise formalization of GB theories is a natural one. This has been done, for instance, by Johnson [Joh89] and Stabler [Sta92] using first-order logic (or the Horn-clause fragment of first-order logic) and by Kracht [Kraar] using a fragment of dynamic logic. What distinguishes the formalization we discuss is the fact that it is carried out in a language which can only define strongly context-free sets. The fact that the formalization is possible, then, establishes a relatively strong language complexity result for the theory we capture.

We have, then, two conflicting criteria for our language. It must be expressive enough to capture the relationships that define the trees licensed by the theory, but it must be restricted sufficiently to be no more expressive than Context-Free Grammars. In keeping with the first of these our language is intended to support, as transparently as possible, the kinds of reasoning about trees typical of linguistic applications. It includes binary predicates for the usual structural relationships between the nodes in the trees—parent (immediate domination), domination (reflexive), proper domination (irreflexive), left-of (linear precedence) and equality. In addition, it includes an arbitrary array of monadic predicate constants—constants naming specific subsets of the nodes in the tree. These can be thought of as atomic labels. The formula NP(x), for instance, is true at every node labeled NP. It includes, also, a similar array of individual constants—constants naming specific individuals in the tree—although these prove to be of limited usefulness. There are two sorts of variables as well—those that range over nodes in the tree and those that range over arbitrary subsets of those nodes (thus this is is monadic second-order language). Crucially, though, this is all the language includes. By restricting ourselves to this language we restrict ourselves to working with properties that can be expressed in terms of these basic predicates.

To be precise, the actual language we use in a given situation depends on the sets of constants in use in that context. We are concerned then with a family of languages, parameterized by the sets of individual and set constants they employ.

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

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

$\approx$ — *equality predicate,*

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

We use infix notation for the fixed predicate constants $\triangleleft$, $\triangleleft^*$, $\triangleleft^+$, $\prec$, and $\approx$. We use lower-case for individual variables and constants, and upper-case for set variables and predicate constants. 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.

Truth, for these languages, is defined relative to a specific class of models. The basic models are just ordinary structures interpreting the individual and predicate constants.

Definition 2 *A* **model** *for the language $L_{K,P}$ 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 universe,
$\mathcal{I}$ 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.

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 $\langle \mathcal{U}, \mathcal{P}, \mathcal{D}, \mathcal{L} \rangle$.

The intended class of these models are, in essence, labeled *tree domains*. A tree domain is the set of node addresses generated by giving the address ϵ to the root and giving the children of the node at address w addresses (in order, left to right) $w \cdot 0, w \cdot 1, \ldots$, where the centered dot denotes concatenation.[2] 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 $uv \in T \Rightarrow u \in T$, ***TD2*** $ui \in T,\ j < i \Rightarrow uj \in T$.

Every tree domain has a natural interpretation as a model for $L_{\emptyset,\emptyset}$ (which interprets only the fixed predicate symbols.)

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

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:*

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

The structures of interest to us are just those models that are the natural interpretation of a tree domain, augmented with interpretations of additional individual and predicate constants.[3]

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

$$M \models \phi$$

asserts that M models ϕ with any assignment. When ϕ is a sentence (has no unquantified variables) we will usually use this form.

Proper domination is a defined predicate:

$$M \models x \triangleleft^+ y\,[s] \Leftrightarrow M \models x \triangleleft^* y, x \not\approx y\,[s].$$

2.1 Definability in $L^2_{K,P}$

We are interested in the subsets of the class of intended models which are definable in $L^2_{K,P}$ using any sets $\boldsymbol{K}$ and $\boldsymbol{P}$. If Φ is a set of sentences in a language $L^2_{K,P}$, we will use the notation $\mathbf{Mod}(\Phi)$ to denote the set of *trees*, i.e., intended models, that satisfy all of the sentences in Φ. We are interested, then, in the sets of trees that are $\mathbf{Mod}(\Phi)$ for some such Φ. In developing our definitions we can use individual and monadic predicates freely (since $\boldsymbol{K}$ and $\boldsymbol{P}$ can always be taken to be the sets that actually occur in our definitions) and we can quantify over individuals and sets of individuals. We will also use non-monadic predicates and even higher-order predicates, e.g., properties of subsets, but only those that can be *explicitly* defined, that is, those which can be eliminated by a simple syntactic replacement of the predicate by its definition.

This use of explicitly defined predicates is crucial to the transparency of definitions in $L^2_{K,P}$. We might, for instance, define a simplified version of government in three steps:

$$\begin{aligned}
\text{Branches}(x) &\leftrightarrow (\exists y, z)[x \triangleleft y \wedge x \triangleleft z \wedge y \not\approx z] \\
\text{C-Command}(x, y) &\equiv \neg x \triangleleft^* y \wedge \neg y \triangleleft^* x \wedge (\forall z)[(z \triangleleft^+ x \wedge \text{Branches}(z)) \rightarrow z \triangleleft^+ y] \\
\text{Governs}(x, y) &\equiv \text{C-Commands}(x, y) \wedge \\
&\quad\ \neg(\exists z)[\text{Barrier}(z) \wedge z \triangleleft^+ y \wedge \neg z \triangleleft^+ x],
\end{aligned}$$

in words, x governs y iff it c-commands y and no barrier intervenes between them. It c-commands y iff neither x nor y dominates the other and every branching node that properly

[3] A partial axiomatization of this class of models is given in [Rog94].

dominates x also properly dominates y. Branches(x) is just a monadic predicate; it is within the language of $L^2_{K,P}$ (for suitable P) and its definition is simply a biconditional $L^2_{K,P}$ formula. In contrast, C-Command and Governs are non-monadic and do not occur in $L^2_{K,P}$. Their definitions, however, are ultimately in terms of monadic predicates and the fixed predicates (parent, etc.) only. One can replace each of their occurrences in a formula with the right hand side of their definitions and eventually derive a formula that *is* in $L^2_{K,P}$. We will reserve the use of $\equiv$ (in contrast to $\leftrightarrow$) for explicit definitions of non-monadic predicates.

Definitions can also use predicates expressing properties of sets and relations between sets, as long as those properties can be explicitly defined. The subset relation, for instance can be defined:

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

We can also capture the stronger notion of one set being partitioned by a collection of others:

$$\text{Partition}(\vec{X}, Y) \equiv (\forall x)\left[\left(Y(x) \leftrightarrow \bigvee_{X \in \vec{X}} X(x)\right) \wedge \bigwedge_{X \in \vec{X}} \left[X(x) \rightarrow \bigwedge_{Z \in \vec{X} \setminus \{X\}} \neg Z(x)\right]\right].$$

Here $\vec{X}$ is a some sequence of set variables and $\bigvee_{X \in \vec{X}} X(x)$ is shorthand for the disjunction $X_0(x) \vee X_1(x) \cdots$ for all X_i in $\vec{X}$, etc. There is a distinct instance of Partiton for each sequence $\vec{X}$, although we can ignore distinctions between sequences of the same length. Finally, we note that finiteness is a definable property of subsets in our intended models. This follows from the fact that these models are linearly ordered by the *lexicographic order* relation:

$$x \trianglelefteq y \equiv x \triangleleft^* y \vee x \prec y.$$

and that every non-empty subset of such a model has a least element with respect to that order. A set of nodes, then, is finite iff each of its non-empty subsets has an upper-bound with respect to lexicographic order as well.

$$\text{Finite}(X) \equiv (\forall Y)[(\text{Subset}(Y, X) \wedge (\exists x)[Y(x)]) \rightarrow (\exists x)[Y(x) \wedge (\forall y)[Y(y) \rightarrow y \trianglelefteq x]]].$$

These three second-order relations will play a role in the next section.

3 Characterizing the Local Sets

We can now give an example of a class of sets of trees that is definable in $L^2_{K,P}$—the local sets (i.e., the sets of derivation trees generated by Context-Free Grammars). The idea behind the definition is simple. Given an arbitrary Context-Free Grammar, we can treat its terminal and non-terminal symbols as monadic predicate constants. The productions of the grammar, then, relate the label of a node to the number and labels of its children. If the set of productions for a non-terminal A, for instance, is

$$\text{A} \longrightarrow \text{Bc} \mid \text{AB} \mid \text{d}$$

we can translate this as

$$\begin{array}{ll}(\forall x)[\mathrm{A}(x) \rightarrow (& (\exists y_1, y_2)[\mathrm{Children}(x, y_1, y_2) \wedge \mathrm{B}(y_1) \wedge \mathrm{c}(y_2)] \vee \\ & (\exists y_1, y_2)[\mathrm{Children}(x, y_1, y_2) \wedge \mathrm{A}(y_1) \wedge \mathrm{B}(y_2)] \vee \\ & (\exists y_1)[\mathrm{Children}(x, y_1) \wedge \mathrm{d}(y_1)] \quad),\end{array}$$

where

$$\begin{array}{rcl}\mathrm{Children}(x, y_1, \ldots, y_n) & \equiv & \bigwedge_{i \leq n}[x \triangleleft y_i] \wedge \bigwedge_{i<j \leq n}[y_i \prec y_j] \wedge \\ & & (\forall z)[x \triangleleft z \rightarrow \bigvee_{i \leq n}[z \approx y_i]].\end{array}$$

We can collect such translations of all the productions of the grammar together with sentences requiring nodes labeled with terminal symbols to have no children, requiring the root to be labeled with the start symbol, requiring the sets of nodes labeled with the terminal and non-terminal symbols to partition the set of all nodes in the tree, and requiring that set of nodes to be finite. It is easy to show that the models of this set of sentences are all and only the derivation trees of the grammar.[4] In this way we get the first half of our characterization of the local sets.

Theorem 1 *The set of derivation trees generated by an arbitrary Context-Free Grammar is definable in* $L^2_{K,P}$.

It is, perhaps, not surprising that we can define the local sets with $L^2_{K,P}$. This is superficially quite a powerful language, allowing, as it does, a certain amount of second-order quantification. It is maybe more remarkable that, modulo a projection, the *only* sets of finite trees (with bounded branching) that are definable in $L^2_{K,P}$ are the local sets.

Theorem 2 *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.*

The proof hinges on the fact that one can translate formulae in $L^2_{K,P}$ into the language of SnS—the monadic second-order theory of multiple successor functions. This is the monadic second-order theory of the structure

$$\mathcal{N}_n \stackrel{\mathrm{def}}{=} \langle T_n, \triangleleft^*, \trianglelefteq, \mathrm{r}_i \rangle_{i<n},$$

a generalization of the natural numbers with successor and less-than. The universe, T_n, is the complete n-branching tree domain. The relation $\triangleleft^*$ is domination, $\trianglelefteq$ is lexicographic order, and the functions r_i are the successor functions, each taking nodes into their i^{th} child ($w \mapsto wi$). Rabin [Rab69] showed that SnS is decidable for any $n \leq \omega$. One way of understanding his proof is via the observation that satisfying assignments for a formula $\phi(\vec{X})$, with free variables[5] among $\vec{X}$ can be understood as trees labeled with (subsets of) the variables in $\vec{X}$. A node is in the set assigned to X_i in $\vec{X}$ iff it is labeled with X_i. Rabin showed that, for any $\phi(\vec{X})$ in the language of SnS, the set of trees encoding the satisfying assignments for $\phi(\vec{X})$ in $\mathcal{N}_n$ is accepted by a particular type of finite-state automaton on infinite trees. We say that the set is *Rabin recognizable.* He goes on to show that emptiness

[4] A more complete proof is given in [Rog94].

[5] We will assume, for simplicity, that only set variables occur free. Since individual variables can be re-interpreted as variables ranging over singleton sets, this is without loss of generality.

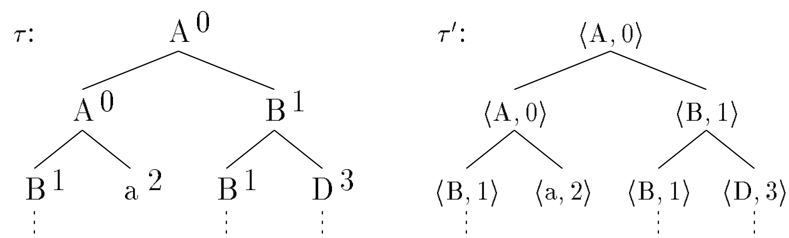

Figure 1: Proof of Theorem 2

of these sets is decidable. It follows that satisfiability of these formulae, and hence the theory SnS, is decidable.

For us, the key point is the fact that the sets encoding satisfying assignments are Rabin recognizable. It is not difficult to exhibit a syntactic transformation which, given any $\psi(\vec{X})$ in $L^2_{K,P}$, produces a formula $\phi(X_U, \vec{X}_P, \vec{X})$ in the language of SnS, where X_U is a new variable and $\vec{X}_P$ is a sequence of new variables (one for each of the finitely many predicates in $\boldsymbol{P}$ that occur in ψ) such that,

$$\mathcal{N}_n \models \phi[A_U, \vec{A}_P, \vec{A}]$$

iff

$$\left\langle A_U, \mathcal{P}^{A_U}, \mathcal{D}^{A_U}, \mathcal{L}^{A_U}, \vec{A}_P \right\rangle \models \psi[\vec{A}],$$

that is, the set A_U and the sequences of sets $\vec{A}_P$ and $\vec{A}$ form a satisfying assignment for ϕ in $\mathcal{N}_n$ iff the structure consisting of the universe A_U along with the natural interpretation of $\triangleleft$, $\triangleleft^*$, and $\prec$ on A_U, and the sets $\vec{A}_P$, satisfies ψ with the assignment taking $\vec{X}$ into $\vec{A}$. It follows that a set of trees is definable in $L^2_{K,P}$ iff they are Rabin recognizable.

If we restrict our attention to sets of finite trees, we can take Rabin's automata to be ordinary finite-state automata over finite trees [GS84], that is, the sets of finite trees that are definable in $L^2_{K,P}$ are simply *recognizable*. One can think of these automata as traversing the tree, top down, assigning states to the children of a node on the basis of a transition function that depends on the state of the node, its label, and the position of the child among its siblings. A tree is accepted if it can be labeled by the automaton in such a way that the root is labeled with a start state and the set of states labeling the leaves is one of a set of accepting sets of states. Every set of trees that is accepted in this way is the projection of a local set. To see this,[6] suppose that τ is a tree accepted by a tree automaton. Then there is some assignment of states to the nodes in τ that witnesses this fact. Suppose, for instance, τ is the tree of Figure 1, labeled as shown. Consider the tree τ' in which each node is labeled with a pair consisting of the label from τ and the state assigned to that node. It is easy to show that, given a recognizable set of trees, one can construct a CFG to generate

[6] This proof is evidently originally due to Thatcher [Tha67]. In addition, Theorem 2 is implicit in the proof of a related theorem due to Doner [Don70].

the corresponding set of trees labeled with pairs as in τ'. In the example, for instance, this would include, among others, the productions

$$\begin{array}{rcl} \langle \mathrm{A}, 0\rangle & \longrightarrow & \langle \mathrm{A}, 0\rangle\, \langle \mathrm{B}, 1\rangle \mid \langle \mathrm{B}, 1\rangle\, \langle \mathrm{a}, 2\rangle \mid \cdots \\ \langle \mathrm{B}, 1\rangle & \longrightarrow & \langle \mathrm{B}, 1\rangle\, \langle \mathrm{D}, 3\rangle \mid \cdots \\ & \vdots & \end{array}$$

The original set of trees is then the first projection of the set generated by the CFG.

Together, these two theorems give us our primary result.

Corollary 1 *A set of finite trees with bounded branching is local (modulo projection) iff it is definable in* $L^2_{K,P}$.

4 Non-Definability of Free Indexation

This characterization provides a powerful tool for establishing strong context-freeness of classes of languages that are defined by constraints on the structure of the trees analyzing the strings in the language. If one can show that the constraints defining such a set, or perhaps that any constraints in the class employed by a given formalism, can be defined within $L^2_{K,P}$ then the corresponding language or class of languages is strongly context-free. Much of the value of standard language complexity classes, on the other hand, comes from results that allow one to show that a given language or class of languages is not included in a particular complexity class. Such results are available here as well, in the form of non-definability results for $L^2_{K,P}$. One relatively easy way of establishing such results is by employing the contrapositive of Theorem 2. If one can show that a given predicate, when added to $L^2_{K,P}$ allows definition of known non-CF languages, then clearly that predicate properly extends the power of the language and cannot be definable. In this way, one can show that the predicate $\mathrm{YieldsEq}_P(x, y)$ which holds between two nodes iff the yields of the subtrees rooted at those nodes are labeled identically wrt P is not definable in $L^2_{K,P}$, for if it were one could define the copy language $\{ww \mid w \in (ab)^*\}$.

In this section we will explore an approach that is more difficult but is one of the most general—reduction from the monadic second-order theory of the grid—and will use it to demonstrate non-definability of free-indexation—a mechanism which shows up in a number of modules of GB.

The grid is the structure $G = \left\langle \mathbb{N}^2, \mathrm{O}, \mathrm{r}_0, \mathrm{r}_1 \right\rangle$ where

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

This is the structure of the (discrete) first quadrant. Note the similarity to $\mathcal{N}_2$, the structure of two successor functions. The key distinction is the fact that G satisfies the property

$$(\forall x)[\mathrm{r}_0(\mathrm{r}_1(x)) = \mathrm{r}_1(\mathrm{r}_0(x))],$$

that is, the horizontal successor of the vertical successor of a point is the same as the vertical successor of its horizontal successor. Let $\mathbf{Th}_2(G)$ be the monadic second-order theory of G.

Lewis [Lew79] showed that this theory is undecidable by showing how one could define the set of terminating computations of an arbitrary Turing machine within it.

Now, the monadic second-order theory of any of our intended structures is decidable (by reduction to SnS), as is the monadic second-order theory of any of our intended structures augmented with any predicate that is definable in $L^2_{K,P}$ (since we can reduce this to the theory of the original structure via that definition). Our approach to showing that a predicate is not definable in $L^2_{K,P}$ is to show that the theory of one of our structures augmented with that predicate is not decidable. In particular, we will show that the theory of such a structure includes an undecidable fragment of the monadic second-order theory of the grid.

Our focus, in this section, is the mechanism known as *free-indexation*. In the Government and Binding Theory framework this is the mechanism that is generally assumed to mediate issues like agreement, co-reference of nominals, and identification of moved elements with their traces. In its most general form this operates by assigning indices to the nodes of the tree randomly and then filtering out those assignments that do not meet various constraints on agreement, co-reference, etc. In essence, the indexation is an equivalence relation, one that distinguishes unboundedly many equivalence classes among the nodes of the tree. That is, each value of the index identifies an equivalence class and there is 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 relationships between the individuals in these classes.

To see that we cannot define such equivalence relations in $L^2_{K,P}$, consider the class of structures

$$\mathcal{T}_{\mathrm{CI}} = \langle T_2, \mathcal{P}_2, \mathcal{D}_2, \mathcal{L}_2, \mathrm{CI} \rangle \,,$$

where T_2 is the complete binary-branching tree domain, $\mathcal{P}_2$, $\mathcal{D}_2$, and $\mathcal{L}_2$ are the natural interpretations of parent, domination, and left-of on that domain, and CI is any arbitrary equivalence relation. Let S2S+CI be the monadic second-order theory of this class of structures. Our claim is that this is an undecidable theory.[7]

Theorem 3 *S2S+CI is not decidable.*

Lewis's proof of the non-decidability of $\mathbf{Th}_2(G)$ is based on a construction that takes any given Turning Machine M into a formula $\phi_M(\vec{P})$ such that $G \models (\exists \vec{P})[\phi_M(\vec{P})]$ iff M halts (when started, say, on the empty tape). The idea behind our proof of the non-decidability of S2S+CI is that there is a natural correspondence between points in T_2 and those in $\mathbb{N}^2$ that is induced by interpreting node addresses in T_2 as paths (non-decreasing in both x and y) from the origin in $\mathbb{N}^2$. Of course, in general, there will be many points in T_2 that correspond to the same point in $\mathbb{N}^2$, but we can restrict the interpretation of CI in such a way that all points in T_2 that correspond to the same point in $\mathbb{N}^2$ will be co-indexed. We then restrict the interpretation of the variables in $\vec{P}$ in such a way that it does not break the classes of CI. In more typically linguistic terms, we require co-indexed nodes to agree on the features in $\vec{P}$.

[7] Since the property of being an equivalence relation—being reflexive, symmetric, and transitive—is definable in $L^2_{K,P}$, our result is one way of showing that $\mathcal{N}_2$ augmented with a single arbitrary binary relation has a non-decidable monadic second-order theory.

The formula $\phi_M(\vec{P})$ of Lewis' proof involves only the constant O, the successor functions r_0 and r_1, some set of (bound) individual variables, the (free) monadic predicate variables in $\vec{P}$, and the logical connectives.

Let

$$\begin{aligned} \mathrm{O}(x) &\leftrightarrow (\forall y)[y \triangleleft^* x \rightarrow y \approx x] \\ \mathrm{r}_0(x, y) &\equiv x \triangleleft y \wedge (\forall z)[x \triangleleft z \rightarrow z \not\prec y] \\ \mathrm{r}_1(x, y) &\equiv x \triangleleft y \wedge (\forall z)[x \triangleleft z \rightarrow y \not\prec z]. \end{aligned}$$

Then $\mathrm{O}(x)$ is true only at the root, $\mathrm{r}_0(x, y)$ is true iff y is the leftmost child of x and $\mathrm{r}_1(x, y)$ is true iff y is the rightmost child of x. These translations are sufficient for us to translate $\phi_M(\vec{P})$ into a formula $\psi_M(\vec{P})$ that, when combined with an axiom $\Phi_G(\vec{P})$ constraining the interpretation of CI and $\vec{P}$ as sketched above, will be satisfiable by a model in the class $\mathcal{T}_{\mathrm{CI}}$ iff $\phi_M(\vec{P})$ is satisfied by G. That is:

$$\text{There exists } T \in \mathcal{T}_{\mathrm{CI}} \text{ such that } T \models (\exists \vec{P})[\psi_M(\vec{P}) \wedge \Phi_G(\vec{P})]$$

iff

$$G \models (\exists \vec{P})[\phi_M(\vec{P})].$$

This in turn implies that

$$(\exists \vec{P})[\phi_M(\vec{P})] \in \mathbf{Th}_2(G) \quad \text{iff} \quad \neg(\exists \vec{P})[\psi_M(\vec{P}) \wedge \Phi_G(\vec{P})] \notin \mathrm{S2S} + \mathrm{CI}.$$

Decidability of S2S+CI, then, would imply decidability of the halting problem.

It remains only to define $\Phi_G(\vec{P})$. Let

$$\begin{aligned}
&\Phi_G(\vec{P}) \equiv \\
&\quad (\forall x, y)[\quad \mathrm{CI}(x, y) \leftarrow (\ x \approx y \quad \vee && \text{(1a)} \\
&\qquad\qquad \text{—}x \textbf{ and } y \textbf{ are equal or} \\
&\qquad\qquad (\exists x_0, y_0)[\ \mathrm{CI}(x_0, y_0) \wedge \\
&\qquad\qquad\quad (\ (\mathrm{r}_0(x_0, x) \wedge \mathrm{r}_0(y_0, y)) \vee \\
&\qquad\qquad\quad\ \ (\mathrm{r}_1(x_0, x) \wedge \mathrm{r}_1(y_0, y))\ \ )] \quad \vee \\
&\qquad\qquad \text{—}x \textbf{ and } y \textbf{ are both left-children or both} \\
&\qquad\qquad\ \ \textbf{right-children of co-indexed nodes or} \\
&\qquad\qquad (\exists x_0, y_0, x_1, y_1)[\ \mathrm{CI}(x_0, y_0) \wedge \\
&\qquad\qquad\quad \mathrm{r}_0(x_0, x_1) \wedge \mathrm{r}_1(x_1, x) \wedge \\
&\qquad\qquad\quad \mathrm{r}_1(y_0, y_1) \wedge \mathrm{r}_0(y_1, y) \quad] \\
&\qquad\qquad \text{—}x \textbf{ is the right-child of the left-child} \\
&\qquad\qquad\ \ \textbf{and } y \textbf{ is the left-child of the right-child} \\
&\qquad\qquad\ \ \textbf{of co-indexed nodes, or v.v.} \\
&\qquad\qquad) \quad \wedge \\
&\qquad \mathrm{CI}(x, y) \rightarrow \mathrm{Agree}_{\vec{P}}(x, y)\ ], && \text{(1b)}
\end{aligned}$$

where

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

This requires that every node is co-indexed with itself, that the left children of co-indexed nodes are co-indexed as are the right children of co-indexed nodes, and that the left child of the right child and right child of the left child of co-indexed nodes are co-indexed. Finally all co-indexed nodes are forced, by $\text{Agree}_{\vec{P}}$, to agree on all predicates in $\vec{P}$. That this is sufficient to carry the reduction of the halting problem to membership in S2S+CI depends on the fact that $\Phi_G(\vec{P})$ forces all points in T_2 equivalent in the sense that they correspond to the same point in G as sketched above, to agree on the predicates in $\vec{P}$. Thus we (roughly) can take the quotient with respect to this equivalence without affecting satisfiability of $\psi_M(\vec{P})$. The resulting structure is isomorphic to G and satisfies $(\exists\vec{P})[\psi_M(\vec{P})]$ iff G satisfies $(\exists\vec{P})[\phi_M(\vec{P})]$. The proof is carried out in detail in [Rog94].

The non-definability of free-indexation is a significant obstacle to capturing GB accounts of language in $L^2_{K,P}$. As it turns out, other constraints employed in GB theories are not generally difficult to define. Our ability to capture these accounts, then, depends directly on the degree to which they necessarily employ free-indexation. The common practice, in GB, is to simply assume co-indexation almost whenever there is a need to identify components of the tree in some way. Unfortunately, we cannot capture directly accounts that are defined in these terms. Rather, we are compelled to restate them without reference to indices. On the other hand, it is not at all clear that accounts that appeal to free-indexation actually require so general a mechanism. On the contrary, it seems that indices are frequently only a conceptually simple way of encoding more complicated, but less general relationships. There has been a tendency, in the more recent GB literature, to avoid free-indexation in favor of these more specific relationships. Chomsky, for instance, comments:

> A theoretical apparatus that takes indices seriously as entities...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. [Cho93, pg. 49, note 52]

This quote comes in the context of a suggestion for a re-interpretation of the standard account of Binding Theory in a manner that avoids use of indices. Rizzi, in [Riz90], motivated by an examination of a wide variety of extraction phenomena, offers a re-interpretation of the Empty Category Principle and the theory of chains that restricts the role of indices to a relatively small class of movements. As we will see in the next section, Rizzi's theory provides us with the foundation we need to capture a largely complete GB account of English in $L^2_{K,P}$. We thus establish that this account licenses a strongly context-free language. It seems noteworthy that GB theorists have been led, by purely linguistic considerations, to precisely the kind of re-interpretation of the theory we require in order to establish our language-theoretic results.

5 Defining Chains

We turn now to an example that is particularly relevant to the issue of capturing a Government and Binding Theory account of English in $L^2_{K,P}$, and in particular capturing it without use of indices. This is our definition of *chains*—the core notion in contemporary GB accounts of movement. Our exposition is intended to be accessible without prior familiarity with GB, although possibly mysterious in some of its details. It will necessarily be

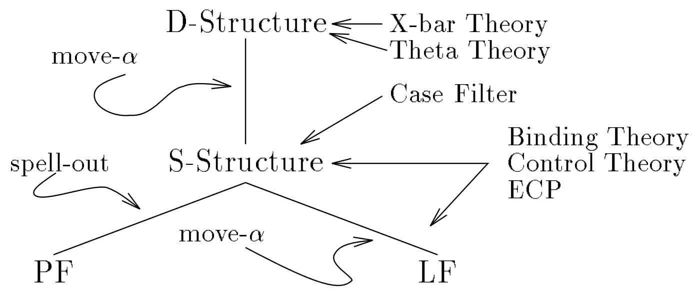

Figure 2: Levels of representation.

somewhat meager both in the details of the definition and in the details of the underlying theory. A more complete treatment can be found in [Rog94].

5.1 Identifying Antecedents of Traces

Government and Binding Theory analyzes sentences with four distinct syntactic representations which are related by the general transformation *move-α*. These are *D-Structure*—corresponding to the deep-structure of earlier transformational theories, *S-Structure*—roughly corresponding to the surface-structure of those theories, *Phonetic Form*—the actual phonetic structure of the sentence, and *Logical Form*—a more or less direct representation of the sentence's semantic content. The principles embodying a GB theory of language are collected into modules which apply at various levels of this analysis. The principles we capture include basic X-bar Theory, Theta Theory, the Case Filter, Binding Theory, Control Theory and various constraints on movement, in particular the Empty Category Principle. In this section we focus on the Empty Category Principle and the definition of *chains*.

As we noted in the introduction, we prefer to regard GB theories as a set of constraints on structures rather than a mechanism for constructing them. We take this a step further by assuming that those constraints apply to a single tree which includes S-Structure and D-Structure as submodels,[8] rather than having some constraints apply to one structure, others to the other, and others still to the relationship between them. In this view, D-Structure and move-α are best understood as perspicuous means of stating constraints which are obscured in a single-level representation (see, for instance, Koster [Kos87] and Brody [Bro93]).[9] One argument against such a view is that in some cases (such as head-raising) chains formed by one movement can be disrupted by subsequent movement. Indeed, representational accounts, such as ours, frequently appeal to a notion of *reconstruction*—effectively derivation in reverse—to resolve such difficulties. In fact, at least if one can employ indices to identify the elements of chains, there is no need for such a retreat. Even limiting oneself to the language of $L^2_{K,P}$, if one restricts attention to languages, like English,

[8] While we don't treat Logical Form, there is no reason this cannot be incorporated into our structures in much the same way.

[9] It is interesting that Johnson, in [Joh89] initially defines all four levels of structure, but then, through a series of standard program transformations, optimizes away everything except PF and LF.

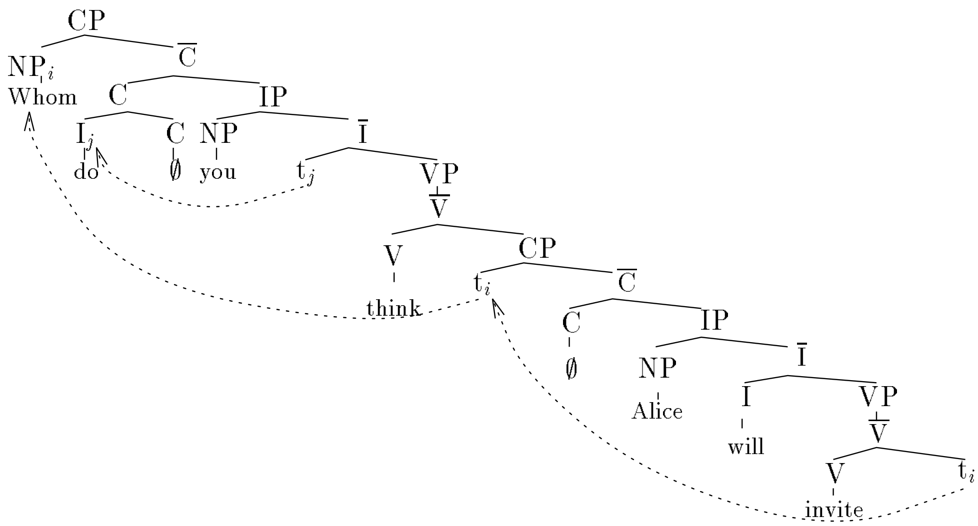

Figure 3: Extraction from the object, S-Structure.

in which head-movement is strictly limited, it is possible to get a purely declarative (and reasonably clear) account of the issues usually treated by reconstruction. Details of such an account are given in [Rog94].

Figure 3 gives the S-Structure of a more or less typical GB analysis of the sentence:

(1) Whom do you think Alice will invite.

In the D-Structure (Figure 4) the element carrying the inflection is positioned between the subject and the predicate and *Whom* is in its standard position as the object of *invite*. Move-α transforms this structure by cutting out the subtrees rooted at I_j and NP_i, leaving phonetically empty traces (t_j and t_i), and re-attaching them a higher positions in the tree. In the case of *Whom* the movement occurs in two steps, with traces being left at each intermediate position. The original position of the moved element is referred to as the *base* position, and its final resting place is the *target* position. The moved element is identified with its traces by co-indexation. Together, an element and the traces co-indexed with it form a *chain*. Chains can be broken up into a sequence of *links* each consisting of a trace and its *antecedent*—the next higher element of the chain.

The fundamental issue we must address in defining chains within $L^2_{K,P}$ is how to identify the antecedent of a trace without reference to indices. Our key idea is that, if we can limit the portion of the tree in which an antecedent can occur, then we can possibly bound the number of potential antecedents a trace may have. Such a bound would suffice since, while we cannot capture indexations with an unbounded range of index, we can capture any indexation in which there is a constant bound on the total number of distinct indices.

In the standard GB account of movement, that of Barriers [Cho86], there are two principles that tend to bound the length of links. The first is *n-subjacency*, which, roughly,

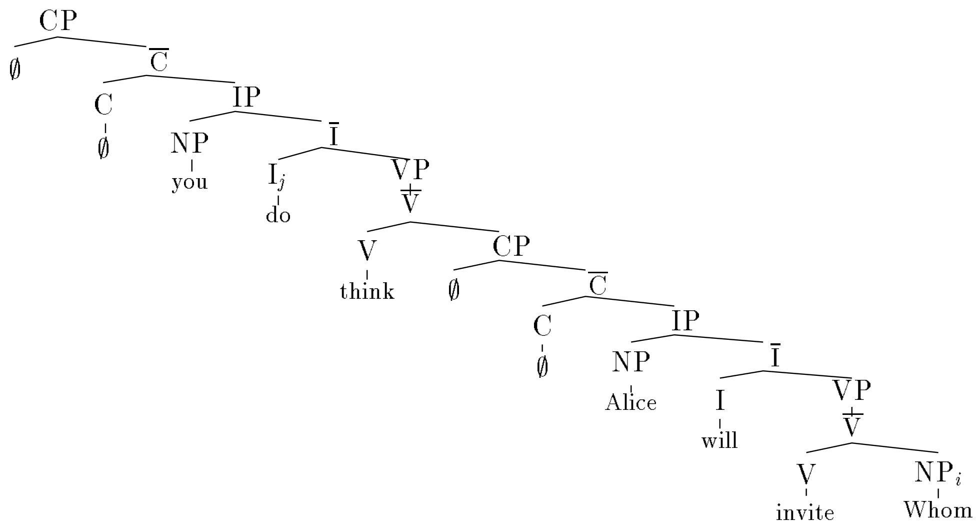

Figure 4: Extraction from the object, D-Structure.

limits the number of phrasal boundaries that a link can cross. This is exactly the kind of constraint we need. Unfortunately it is responsible only for weak effects; there are many sentences that violate n-subjacency that are only of degraded acceptability rather than outright ungrammatical. The second principle that might do is the *Empty Category Principle*. This puts specific constraints on the structural relationship between a trace and its antecedent. Indices, however, play a significant role in Chomsky's formulation of this principle.

There is a formulation of ECP, due to Rizzi and based on his notion of *Relativized Minimality* [Riz90], in which the role of indexation is largely eliminated. In Rizzi's theory, this is a conjunctive principle with two components, a Formal Licensing requirement and an Identification requirement:

ECP (Rizzi):

- A non-pronominal empty category must be properly head-governed. (Formal Licensing)
- Operators must be identified with their variables. (Identification)

We are interested in the identification requirement, which, incidently, is responsible for most of the effects attributed to ECP in the Barriers account. This constraint requires every trace (variable) to be identified with its target (operator). This can be done in one of two ways, either by a particular class of index, the referential indices, or by a sequence of *antecedent-government* links. In the latter case the role of indices in identifying chains can be taken over by the antecedent-government relation.

To a first approximation, government is simply a relation between an element and those elements occurring in a specifically limited region of the tree dominated by the phrase in

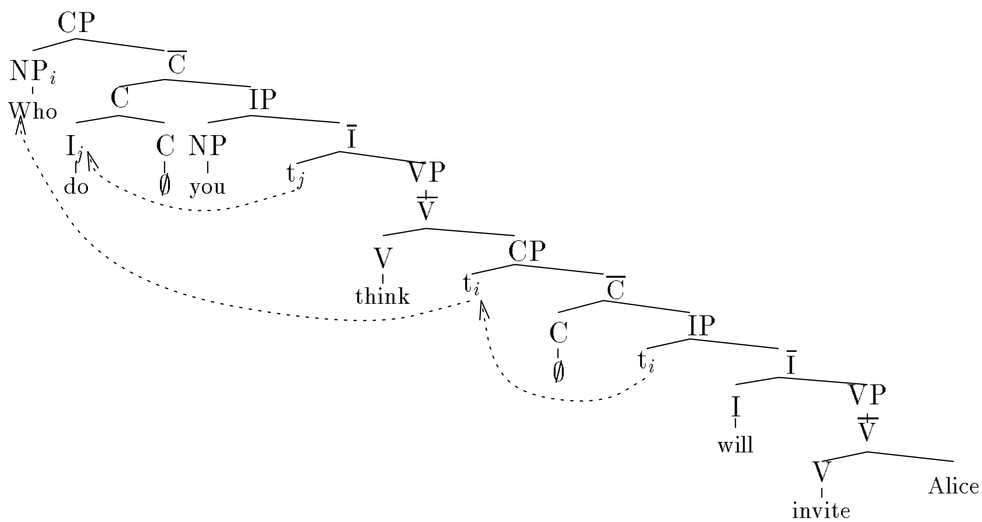

Figure 5: Extraction from the subject.

which that element (the governor) occurs. Its definition has three components. First, for the class of government relations we are considering here, the governor must c-command the elements it governs, that is, those elements must be dominated by a sibling of the governor. Second, there must be no intervening barrier. For Rizzi, the notion of barrier is much weaker than it is in the Barriers account. Here, this constraint simply forbids the government relation from crossing certain phrasal boundaries (in particular specifiers, adjuncts and complements of nouns or prepositions). The final component of the government relation requires a governor to be the minimal potential governor of the elements it governs, that is, no potential governor can fall properly between a governor and the elements it governs. There are a range of types of government relations that fall under this general category. In Rizzi's theory only potential governors of the same type count for the minimality requirement. (This is the relativized aspect of his theory.) For antecedent-government there is an additional requirement that the governor be co-indexed with the trace.

Definition 5 *x antecedent-governs y iff*

- *x c-commands y.*
- *No barrier falls between x and y.*
- *Minimality is respected.*
- *x and y are co-indexed.*

As we will see, we can drop the co-indexation requirement on the grounds that, when it exists, the antecedent-governor is unique.

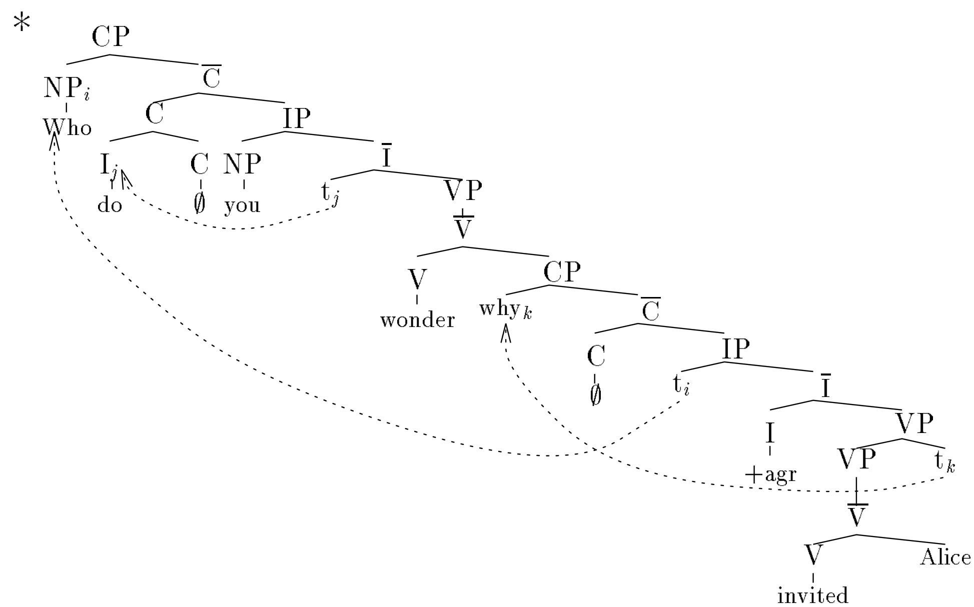

Figure 6: An ECP violation.

As an example of these relationships, consider, in Figure 5, the trace in the *specifier* of the lower IP, that is, the trace of *Who* falling immediately under the IP. The elements c-commanding this trace include the (empty) C, the t_i in the specifier of CP, the V, etc. This is a *Wh-Trace* which means that, by the principles of Binding Theory, its antecedent must fall in a *non-argument* position. In the example, the non-argument positions c-commanding the trace are just the specifiers of the CPs. By minimality, no potential antecedent of the trace beyond the closest specifier of CP can govern it. Thus the only possible antecedent-governor of the trace in question is the trace in the specifier of the lower CP, which is, in fact, its antecedent.

In contrast, if we fill that position with a moved adverbial, as in the example of Figure 6, there is a problem. The element *why* cannot be the antecedent of the trace in the specifier of the lower IP, but it blocks government by all other potential antecedents. Thus the trace t_i cannot be identified with its antecedent, and the sentence is ruled ungrammatical on the grounds that it violates ECP.

In this way, minimality suffices to pick out the unique antecedent of traces in chains that are identified by antecedent-government. But under Rizzi's criteria chains can also be identified by referential indices. These are just indices assigned to elements that receive what are termed *referential* Theta roles. Again to a first approximation, we can take these simply to be elements that are the objects of verbs. In Figure 6 *Who* is extracted from the embedded subject. If we return to our original example, in which we extract from the object, we find that filling the specifier of the lower CP with a moved adverbial (Figure 7) has a less dramatic effect. While antecedent government of the trace in the complement of

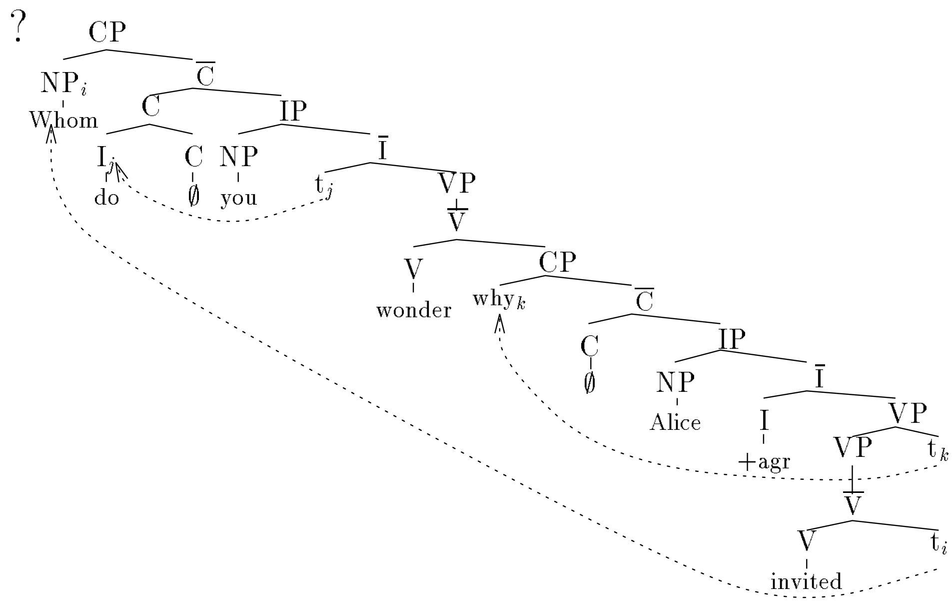

Figure 7: A 1-subjacency violation.

the lower VP is blocked, that trace can now be identified with its target by the referential index they share. The fact that this example is not judged to be as bad as the example from Figure 6 is attributed, then, to the fact that it is only a 1-subjacency violation rather than an ECP violation.

In general, we could be forced to resort to a mechanism equivalent to indexation in order to distinguish such referential chains. It turns out, however, that in English, at least, chains of this type do not overlap. Manzini [Man92], in fact, argues for an account of $\overline{\text{A}}$-movement (movements, like these we have been considering, to non-argument positions) which implies that no more than two such chains—one referential and one non-referential—may ever overlap. Thus, we need to identify only a single referential antecedent in any single context.

5.2 Defining Antecedent-Government, Links, and Chains

Relativized Minimality theory distinguishes a number of distinct varieties of antecedent-government, one for each class of movement. We look at one representative case $\overline{\text{A}}$-antecedent-government. This is defined, in $L^2_{K,P}$ as follows:

$\overline{\text{A}}$-Antecedent-Governs$(x, y) \equiv$

$\quad \neg\text{A-pos}(x) \wedge \text{C-Commands}(x, y) \wedge \text{T.Eq}(x, y) \wedge$

$\qquad$ —x **is a potential antecedent in an $\overline{\text{A}}$-position**

$\quad \neg(\exists z)[\text{Intervening-Barrier}(z, x, y)] \wedge$

—**no barrier intervenes**

$\neg(\exists z)[\text{Spec}(z) \wedge \neg\text{A-pos}(z) \wedge$
$\text{C-Commands}(z, x) \wedge \text{Intervenes}(z, x, y)]$

—**minimality is respected**

In words, this says simply that x is an $\overline{\text{A}}$-antecedent-governor of y iff x is in a non-argument ($\overline{\text{A}}$) position, it c-commands y, no barrier intervenes between x and y, and no non-argument specifier falls between x and y. The actual definitions of A-Pos, T.Eq, Intervening-Barrier, Spec, and Intervenes is unimportant here. The predicate T.Eq is used to check the compatibility of the features of the trace with those of its antecedent.

Using this, we can define the link relation.

$\overline{\text{A}}\text{-}\overline{\text{Ref}}\text{-Link}(x, y) \equiv$

$\overline{\text{A}}\text{-Antecedent-Governs}(x, y) \wedge \neg\text{Ref}(x) \wedge \neg\text{Ref}(y) \wedge$
$\text{Bar2}(x) \wedge (\neg\text{Target}(x) \vee \text{Spec}(x)) \wedge$

—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**

This is just antecedent-government with certain additional configurational requirements. We can extend the notion of links based on Rizzi's antecedent-government to include antecedents and traces that Rizzi identifies with a referential index (which we refer to as $\overline{\text{A}}$-referential links), and links formed by rightward movement. This gives us five distinct link relations. As they are mutually exclusive, we can take their disjunction to form a single link relation which must be satisfied by every trace and its antecedent.

$$\begin{aligned}\text{Link}(x, y) \quad \equiv \quad & \text{A-Link}(x, y) \vee \overline{\text{A}}\text{-}\overline{\text{Ref}}\text{-Link}(x, y) \vee \\ & \overline{\text{A}}\text{-Ref-Link}(x, y) \vee \text{X}^0\text{-Link}(x, y) \vee \\ & \text{Right-Link}(x, y)\end{aligned}$$

The idea, now, is to define chains as any set of nodes that are linearly ordered by Link. Before we can do this, though, we have one more issue to resolve. The problem is that, while we can identify a unique antecedent for each trace, nothing assures us that there will be a unique trace for each antecedent, that is, nothing prevents us from identifying the same node as the antecedent of more than one trace. As an example, we might license the tree in Figure 8. This is the conflation of two sentences:

(2) a. Who$_i$ has t$_i$ told you Alice invited him.

b. Who$_i$ has Alice told you t$_i$ t$_i$ invited him.

In the first we have extracted *Who* from the subject of the matrix clause and in the second we have extracted it from the subject of the embedded clause. We can find a link relation

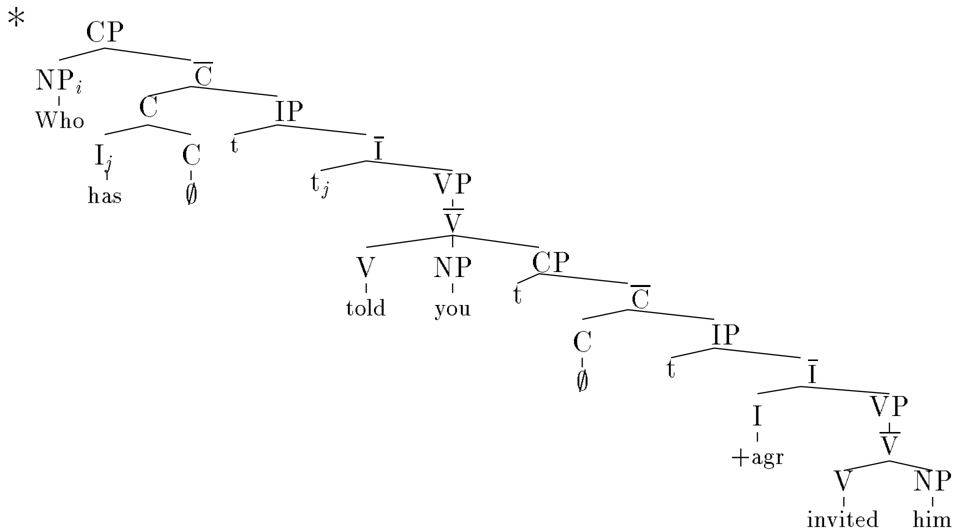

Figure 8: Conflated chains.

between *Who* and the trace in the specifier of the matrix IP and a link relation between *Who* and the trace in the specifier of the embedded CP, but clearly it cannot have moved from both positions.

We rule out such structures by requiring that chains not only be linearly ordered by Link, but that they are also closed under the link relation, that is, every chain includes every node that is related by Link to any node in the chain. Trees like the the one in Figure 8 are ruled out on the grounds that any chain that contains either of the traces in question must include both of them, and will therefore not be linearly ordered.

Formalizing this, we get:

$$
\begin{aligned}
&\text{Chain}(X) \equiv \\
&\quad (\exists! x)[X(x) \wedge \text{Target}(x)] \wedge (\exists! x)[X(x) \wedge \text{Base}(x)] \wedge \\
&\qquad \text{—}X \textbf{ contains exactly one Target and one Base} \\
&\quad (\forall x)[X(x) \wedge \neg\text{Target}(x) \rightarrow (\exists! y)[X(y) \wedge \text{Link}(y,x)]] \quad \wedge \\
&\qquad \text{—}\textbf{All non-Target have a unique antecedent in } X \\
&\quad (\forall x)[X(x) \wedge \neg\text{Base}(x) \rightarrow (\exists! y)[X(y) \wedge \text{Link}(x,y)]] \quad \wedge \\
&\qquad \text{—}\textbf{All non-Base have a unique successor in } X \\
&\quad (\forall x,y)[X(x) \wedge (\text{Link}(x,y) \vee \text{Link}(y,x)) \rightarrow X(y)] \\
&\qquad \text{—}X \textbf{ is closed wrt the Link relation}.
\end{aligned}
$$

5.3 Defining the ECP

We can now capture Rizzi's version of the Empty Category Principle:

Licensing

$$(\forall x)[\text{Trace}(x) \rightarrow (\text{Bar0}(x) \vee (\exists y)[\text{Proper-Head-Governs}(y, x)])]$$

Identification

$$(\forall x)[\text{Trace}(x) \rightarrow (\exists X)[\text{Chain}(X) \wedge X(x)]]$$

Note, in particular, that in our definition the identification requirement is reduced simply to a requirement that every trace is a member of some well-formed chain. As we admit the notion of *trivial* chains—chains with a single element, formed by zero movements—we can generalize this to a global requirement that every element of the tree is a member of a (possibly trivial) well-formed chain.

Identification (Generalized)

$$(\forall x)(\exists X)[\text{Chain}(X) \wedge X(x)].$$

Recall that identification is the component of Rizzi's definition that accounts for most of the effects attributed to ECP in the Barrier's account of movement. Thus we have reduced a variety of effects to a single simple global principle. Of course we have paid for this with a complex definition of chains, but much of this complexity lies in the definition of antecedent-government and Rizzi argues, on linguistic grounds, for essentially this definition in any case. It is satisfying that we can recover its added complexity in the form of a greatly simplified ECP.

5.4 Limits of the Definition

The fact that we can exhibit a definition in $L^2_{K,P}$ of the class of trees licensed by a specific GB account of English provides a strong complexity result for that class of trees—it is strongly context-free. We don't, on the other hand, expect this formalization to work for GB theories in general, and, in particular we don't expect it to work for a GB account of Universal Grammar. A more or less typical account of head-raising in Dutch, for instance, is given in Figure 9. This is the type of movement presumed to be responsible for the cross-serial dependencies that form the basis of Shieber's claim that Swiss-German is non-context-free [Shi85]. Bresnan, et al., [BKPZ82] have pointed out that analyses such as these form a non-recognizable set. Consequently, it cannot be possible to capture this account within $L^2_{K,P}$, and, in fact, the definition we give fails to license these structures. Examining why this is the case provides some insight into the kinds of natural properties of linguistic structures that correspond to increased language-theoretic complexity.

In order to rule out the possibility of "forking" chains—of some nodes participating in the licensing of multiple gaps—we have required chains to be maximal in the sense that they include every node that is related by link to any node in the chain. Consequently, we can license overlapping chains only if they are distinguished in some way. The account works for English because we can classify chains in English into a bounded set of types in such a way that no two chains of the same type ever cross. (This fact depends to a great extent on the minimality requirement in the antecedent-government relation.) This property can be stated as a principle:

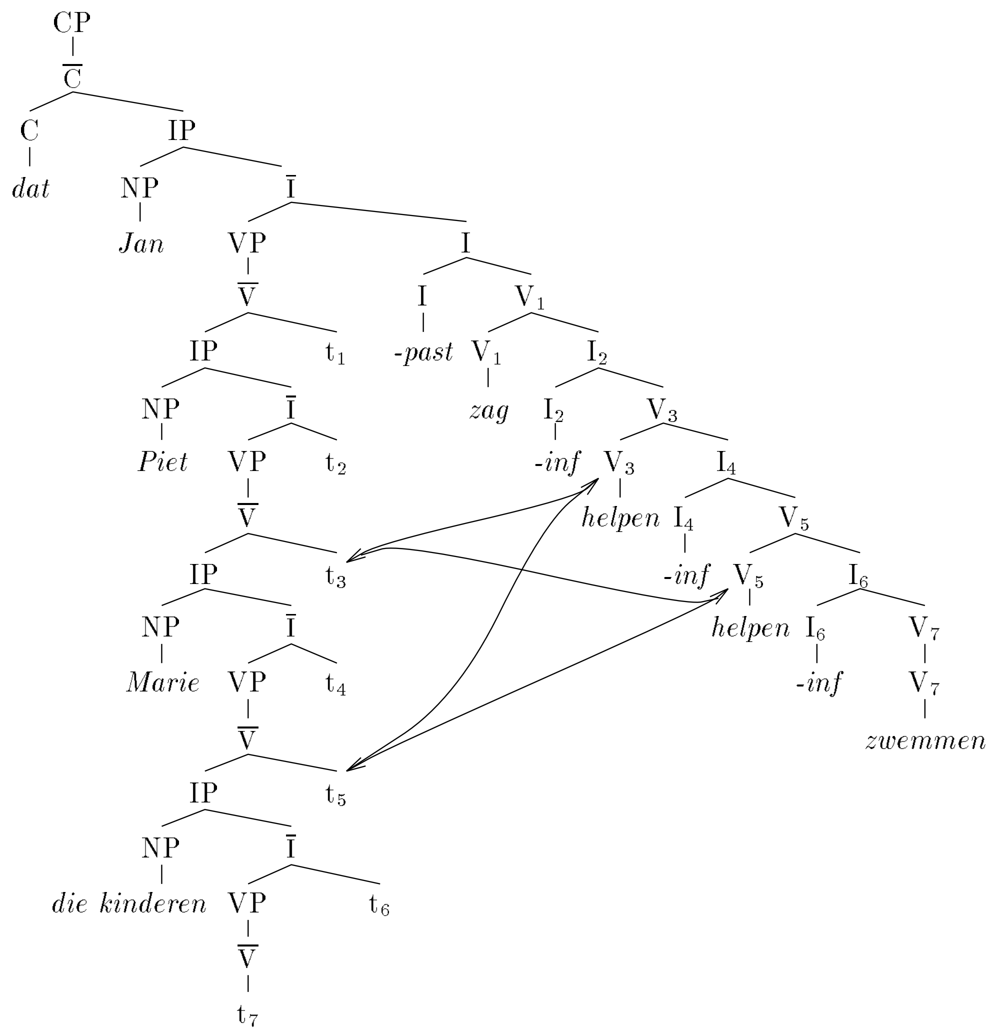

Figure 9: Head-Raising in Dutch

The number of chains which overlap at any single position in the tree is bounded by a constant.

Our approach to chains will work for any account of language that satisfies this principle. Once again, the linguistics literature provides arguments that such bounds exist, at least in some cases. As we have already noted, Manzini's *Locality Theory* [Man92] implies that no more than two $\overline{\text{A}}$-chains ever overlap. Stabler [Sta94] makes the stronger claim that such bounds exist for all linguistically relevant relationships in all languages.

Leaving aside the possibility that it may be possible to account for cross-serial dependencies in Dutch in other ways, we can note that accounts employing structures such as the one in Figure 9 fail to meet the bound on overlapping chains. This is despite the fact that, if one orders the movements bottom-up, each movement meets the strictest conceivable locality constraint—each head moves to the closest possible position (often stated as the *Head Movement Constraint*). The problem is that, even if the movements are ordered in this way, each movement carries the target positions of the prior movements along with it. Thus, in the final structure all chains of head-movement overlap. Given that the number of heads participating in these structures is arbitrary, there can be no *a priori* bound on the number of overlapping chains. Note that in the example the two *helpen* chains ($[\mathrm{V}_3, t_3]$ and $[\mathrm{V}_5, t_5]$) are indistinguishable. Any attempt to form a chain including any of these nodes will be required to include all four and the result will not be linearly ordered.

6 Conclusion

In this paper we have introduced a kind of descriptive complexity result for the strongly Context-Free Languages—a language is strongly context-free iff the set of trees analyzing the syntax of its strings is definable in $L^2_{K,P}$ (modulo a projection). Using this result we have sketched a couple of language complexity results relevant to GB, namely, that free-indexation cannot, in general, be enforced by CFGs, and that a specific GB account of English licenses a strongly context-free language. The first of these results is not likely to come as a surprise to the GB community. The appropriateness of free-indexation as a fundamental component in linguistic theories has been questioned in the more recent GB literature on purely linguistic (rather than complexity theoretic) grounds.

The second result is more surprising. We don't expect it to extend to the whole range of human languages, that is, to any theory of Universal Grammar. Shieber [Shi85] and Miller [Mil91] (to cite two examples) give fairly strong evidence that there are constructions that occur in human languages that are beyond the CFLs, and hence not possible to capture in $L^2_{K,P}$. As expected, our definitions fail for these constructions. The fact that the definition works for English is a consequence of the fact that, in the account of English we capture, it is possible to classify chains into finitely many categories in such a way that no two chains from a given category ever overlap. GB-style analyses of the constructions studied by Shieber and by Miller include positions in which an unbounded number of chains can overlap. Our definition is unable to identify any well-formed chains including these positions; indeed, there is unlikely to be any way to distinguish these chains without the equivalent of unbounded indices.

As it stands, this result speaks only of the particular account of English we capture. The fact that this is context-free says nothing about the nature of human language faculty, since the principle it depends upon is unlikely to be a principle of Universal Grammar. It does, however, raise the prospect of wider results. Extensions of our descriptive complexity result to larger language complexity classes could provide formal restrictions on the principles employed by GB theories that would be sufficient to provide non-trivial generative capacity results for those theories without losing the ability to capture the full range of human language. With such extended characterizations one might establish upper bounds on the complexity of human language in general. The possibility that such results might be obtainable is suggested by the fact that we find numerous cases in which the issues arising in our studies for definability reasons, and ultimately for language complexity reasons, have parallels that arise in the GB literature 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 complexity classes.

References

[Ber84] Robert C. Berwick. Strong generative capacity, weak generative capacity, and modern linguistic theories. *Computational Linguistics*, 10:189–202, 1984.

[BKPZ82] Joan Bresnan, Ronald M. Kaplan, Stanley Peters, and Annie Zaenen. Cross-serial dependencies in Dutch. *Linguistic Inquiry*, 13:613–635, 1982.

[Bro93] Michael Brody. Θ-theory and arguments. *Linguistic Inquiry*, 24(1):1–23, 1993.

[BW84] Robert C. Berwick and Amy S. Weinberg. *The Grammatical Basis of Linguistic Performance*. MIT Press, 1984.

[Cho86] Noam Chomsky. *Barriers*. MIT Press, 1986.

[Cho93] Noam Chomsky. A minimalist program for linguistic theory. In *The View from Building 20*, pages 1–52. MIT Press, 1993.

[Don70] John Doner. Tree acceptors and some of their applications. *Journal of Computer and System Sciences*, 4:406–451, 1970.

[GS84] Ferenc Gécseg and Magnus Steinby. *Tree Automata*. Akadémiai Kiadó, Budapest, 1984.

[Joh89] Mark Johnson. The use of knowledge of language. *Journal of Psycholinguistic Research*, 18(1):105–128, 1989.

[Kos87] Jan Koster. *Domains and Dynasties*. Foris Publications, Dordrecht, Holland, 1987.

[Kraar] Marcus Kracht. Syntactic codes and grammar refinement. *Journal of Logic, Language, and Information*, to appear.

[Lap77] Steven G. Lapointe. Recursiveness and deletion. *Linguistic Analysis*, 3(3):227–265, 1977.

[Lew79] Harry R. Lewis. *Unsolvable Classes of Quantificational Formulas.* Addison-Wesley, 1979.

[Man92] Maria Rita Manzini. *Locality: A Theory and Some of Its Empirical Consequences.* MIT Press, Cambridge, Ma, 1992.

[Mil91] Philip H. Miller. Scandinavian extraction phenomena revisited: Weak and strong generative capacity. *Linguistics and Philosophy*, 14:101–113, 1991.

[PG82] Geoffrey K. Pullum and Gerald Gazdar. Natural language and context-free languages. *Linguistics and Philosophy*, 4:471–504, 1982.

[Pul84] Geoffery K. Pullum. On two recent attempts to show that English is not a CFL. *Computational Linguistics*, 10:182–188, 1984.

[Rab69] Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, July 1969.

[Riz90] Luigi Rizzi. *Relativized Minimality.* MIT Press, 1990.

[Rog94] James Rogers. *Studies in the Logic of Trees with Applications to Grammar Formalisms.* Ph.D. dissertation, Univ. of Delaware, 1994.

[Shi85] Stuart M. Shieber. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8:333–343, 1985.

[Sta92] Edward P. Stabler, Jr. *The Logical Approach to Syntax.* Bradford, 1992.

[Sta94] Edward P. Stabler. The finite connectivity of linguistic structure. In C. Clifton, L. Frazier, and K. Rayner, editors, *Perspectives on Sentence Processing*, chapter 13, pages 303–336. Lawrence Erlbaum, Hillsdale, NJ, 1994.

[Tha67] J. W. Thatcher. Characterizing derivation trees of context-free grammars through a generalization of finite automata theory. *Journal of Computer and System Sciences*, 1:317–322, 1967.