"Grammarless" Phrase Structure Grammar

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

November 3, 1997

Abstract

We sketch an axiomatic reformalization of Generalized Phrase Structure Grammar (GPSG)—a definition purely within the language of mathematical logic of the theory GPSG embodies. While this treatment raises a number of theoretical issues for GPSG, our focus is not the reformalization itself but rather the method we employ. The model-theoretic approach it exemplifies can be seen as a natural step in the evolution of constraint-based theories from their grammar-based antecedents. One goal of this paper is to introduce this approach to a broader audience and to demonstrate its application to an existing theory. As such, it joins a growing literature of similar studies. Prior studies, however, have had a number of weaknesses—they generally offer little in the way of concrete examples of the advantages the approach has to offer, they typically ignore significant portions of the theories they address, and, by fully abstracting away from the notion of grammar mechanism, they largely abandon the possibility of establishing meaningful complexity results. The second goal of the paper is to address these issues. Our thrust is to sketch the reformalization sufficiently to illustrate the way in which it captures Feature Specification Defaults (FSDs) and the Exhaustive Constant Partial Ordering (ECPO) property. Our definition of FSDs is considerably simplified relative to the original formalization and is free of the procedural flavor that has led some to assume that FSDs are inherently dynamic. Our treatment of ECPO uncovers a gap in its definition in the context of partial categories that has heretofore gone unnoticed. We offer these as a demonstration of the kind of insight that a model-theoretic re-interpretation can bring to an existing theory. Further, since these are the types of properties that prior studies in this genre have failed to address, FSDs and ECPO provide a means for us to explore the limitations of these approaches and to offer ways of overcoming them. Finally, the logical framework we employ has a well defined generative capacity—definability in this framework characterizes strong context-freeness in a particular sense. Thus, despite being more abstract than its constraint-based predecessors, the model-theoretic approach, as exemplified here, can offer stronger complexity results than are typically available in the constraint-based framework.

1 Introduction

The origins of generative grammar, almost by definition, are firmly rooted in a procedural notion of grammar—a notion in which the grammar formalism provides a general mechanism for generating or recognizing strings, with the grammar giving a specific instance of that mechanism. Part of the legacy of this approach is the fact that linguistic theories couched in terms of such a formalism are expressed partly by the grammar and partly by the characteristics of the mechanism. That is, some of the properties of natural language that comprise the theory—in particular those that are universal—are consequences of the specific nature of the formalism. In the extreme, one might hope that the grammar formalism embodies exactly the universal characteristics of the human language faculty, that the class of languages definable within the formalism coincides with the class of all (potential) human languages.

Over time, the restrictions built into formalisms have weakened considerably and the role of these restrictions in defining linguistic theories has correspondingly diminished. There are probably a variety of reasons for this, but the most obvious is a desire to express linguistic theories in a more direct way, to use the formalism primarily to fix a language of discourse and its meaning and then to state the linguistic theory explicitly in terms of that language. The grammar formalism, then, provides a class of structures along with a more or less precise way of stating constraints on those structures; the linguistic theory is expressed as a set of constraints that characterize the set of structures correctly analyzing the strings in the language. Among the formalisms taking this approach one can include not only the obvious constraint-based formalisms (LFG, FUG, HPSG, PATR II, etc.), but also the formal framework of GB theories. In all of these approaches, the class of languages definable within the formalism is no longer an issue, except that it must be large enough to include the natural languages as a class. One is primarily concerned only with whether the formalism is adequate to express the generalizations comprising the linguistic theory.

A system of constraints expressing a grammatical theory within one of these formalisms is, in essence, a definition of the properties of the class of structures licensed by the theory couched within a specific formal framework. For the most part, properties of classes of structures that are defined in a formal way like this are the provenance of Model Theory. It's not surprising, then, to find treatments of the meaning of such systems of constraints couched in terms of formal logic [KR86, MR87, KR90, GPC+88, Joh88, Smo89, DVS90, Car92, Kel93, RVS94]. More recently, a number of people have noticed that, at least in some cases, extra-logical mechanisms for combining constraints can be replaced by ordinary logical operations. (See, for instance, [Joh89, Sta92, Cor92, BGMV93, BMV94, Kel93, Rog94, Kra95], and, anticipating all of these, [JP80].) This approach abandons the notion of grammar as a mechanism and, instead, defines a language as a class of more or less ordinary mathematical structures via a linguistic theory expressed in a more or less ordinary logical language.[1]

While this seems to be a natural step in the evolution of formal linguistics, there are a number of potential weaknesses in the model-theoretic approach that the existing studies leave open. First, they tend to concentrate on showing that it is possible to re-interpret grammar formalisms within the model-theoretic framework and are, perhaps, somewhat weak in justifying why one might want to undertake them—in showing what such reinter-

[1] One is tempted to lump these approaches under the rubric of Model-Theoretic Syntax.

pretations have to offer linguistics. Second, by focusing only on the set of structures licensed by the grammatical theories, we will argue, they fail to treat many linguistically significant aspects of those theories. Finally, as with the constraint-based theories, the move to relatively unrestricted formalisms makes it difficult to establish language-theoretic complexity results for these theories. While there is no longer much reason to believe that such results will serve to characterize the human languages, they are still arguably the most practical abstract means of judging the adequacy of grammatical theories, and they still provide evidence about the underlying nature of the human language faculty. The loss of these potentials is regrettable.

In this paper we address these issues within the context of a model-theoretic reinterpretation of Generalized Phrase Structure Grammar (GPSG) [GKPS85]. There is an element of anachronism in this, since GPSG has largely been overshadowed by its successors; there is little current research in GPSG. Nonetheless, it provides an ideal laboratory for exploring these issues for a variety of reasons. In many respects GPSG epitomizes the tradition of expressing linguistic principles as restrictions to the formalism. In GKP&S there is an explicit intent to encode all universal principles of grammar within the formalism itself. At the same time, GPSG is an early exemplar of the declarative approach underlying constraint-based theories. Grammar rules, in GPSG, are interpreted as licensing sets of trees rather than generating them—of selecting the set of well-formed trees from the set of all finite labeled trees rather than building them from a set of simple trees. The grammar, then, is a means of specifying a set of constraints on the structure of trees, which together define the class of well-formed analyses of English utterances. Consequently, issues raised by the encoding of linguistic principles with the formalism are separated, in GPSG, from those having to do with its procedural interpretation.

The framework we employ, $L^2_{K,P}$, a monadic second-order language for reasoning about trees, is particularly well suited for this task. First, while it is superficially quite an expressive language—allowing quantification over arbitrary sets as well as over individuals—it has well defined generative capacity—the sets of finite trees definable in $L^2_{K,P}$ are a slight generalization of the sets of trees licensed by the phrase structure component of GPSG grammars. Thus this logic provides us with a model-theoretic approach to formalizing theories of syntax that has restricted strong generative capacity; we get the advantages of the grammar-based approach with respect to language-theoretic complexity results and get, at the same the time, advantages of the model-theoretic approach with respect to naturalness and clarity in expressing linguistic principles.

Second, because it allows second-order quantification, $L^2_{K,P}$ lets us treat many of those aspects of linguistic theories that have been ignored by prior model-theoretic interpretations. We will look, in particular, at Feature Specification Defaults (FSDs) and the Exhaustive Constant Partial Ordering property (ECPO). While the original formalization of FSDs is actually declarative, it is couched in terms that obscure this fact—so much so that FSDs have been cited as an inherently dynamic aspect of GPSG. The second-order quantification of $L^2_{K,P}$ allows a definition that is considerably simpler and that is clearly independent of the mechanisms that might be employed to enforce the defaults. ECPO is the universal principle justifying the ID/LP format of GPSG. We will argue that it is a higher-order property and cannot show up in first-order theories. We show, however, that it can be captured in $L^2_{K,P}$. Our treatment of FSDs serves to dispel a common misconception about their nature; our

treatment of ECPO illuminates a gap in the theoretical underpinnings of GPSG that has heretofore gone unnoticed. Thus they serve to address the question of what motivates the model-theoretic approach. By focusing on the way restrictions to the grammar are reflected in properties of the structures the grammar licenses, one tends to raise issues about the intended meaning of those restrictions that are not necessarily apparent in their original form. The fact that such treatments can illuminate structural consequences of restricted grammar formalisms that are otherwise obscure is a powerful argument for the benefit of the approach.

1.1 Mathematical Theories and Linguistic Theories

One of the first issues that confronts attempts to interpret existing grammatical theories model-theoretically is the fact that the notion of *theory* in formal logic does not coincide precisely with the notion of *linguistic theory*. On one level a grammar is just a means of specifying a set of structures that encode the syntax of a language or class of languages. On this level, we are interested in the theory of the language in the mathematical sense—the set of all assertions that are true of every structure in the set of well-formed analyses. The grammar is a (generally) finite means of specifying a subset of those assertions that entails the entire theory, i.e., an axiomatization of that set of well-formed structures. Every presentation of a logically equivalent set of axioms has equal status on this level. Indeed, all linguistic theories that agree on ontology and the empirical facts should agree at this level. This is the level to which the usual notions of generative capacity apply. We will refer to this as the Descriptive Level; since we might just as well take any algorithm for enumerating the language to be the grammar, linguistics at this level is essentially descriptive.

But this level is not the sole focus, and arguably not even the primary focus, of linguistic theories. Linguistically, one is concerned with *how* the syntax of the language is defined—with which generalizations about the language are germane and with how those generalizations are expressed. In other words, linguistic theories are largely concerned with the choice of axioms and their manner of presentation. When linguists propose a relationship between simple declarative sentences and their passive form—as a class of transformations, as a set of metarules, or however—they are not *just* making a claim about a closure property of the set of strings in the language, they are also making an assertion about grammatical competence, about the organization of linguistic knowledge. They are not just predicting grammaticality judgments, but they are making a claim about how those judgments are made, the nature of the reasoning underlying them[2]. We will refer to such aspects of linguistic theories as forming the Theoretic Level.[3] Note that equivalence at this level is stronger than the notions of either weak or strong generative capacity. There may be many ways of axiomatizing the same set of structures, each of which misses some of the generalizations captured by the others.[4]

[2] These are issues of what Chomsky terms *explanatory adequacy* [Cho86]. While it is at this level that such issues arise, we are more properly concerned here with the organization of the linguistic theory itself, not necessarily the organization of human linguistic knowledge. The latter is the usual justification of the former, but it is not the only possible justification. We include at the this level all significant aspects of the organization of the grammar, not just those that are psychologically motivated.

[3] We should note that the terms *descriptive* and *theoretic level* are not original with us, going back at least to Bresnan and Kaplan [Bre82]. Our usage, though, is not necessarily consistent with earlier uses.

[4] Of course, these notions are relative. What we refer to as the descriptive level here, when considering

Model-theoretic interpretations of linguistic theories have generally focused on the set of structures licensed by a particular instance of a grammar, that is, on the descriptive level. For instance, the metarules of GPSG—rules that capture generalizations about the phrase-structure rules of the GPSG grammar—are typically treated, in model-theoretic reformalizations, as a notational convenience and are assumed to be "multiplied out." The subject of the interpretation is the set of phrase-structure rules generated by application of the metarules. This is natural. In GPSG a metarule asserts, in essence, that wherever a local tree of a given form is licensed then a local tree of a related form is also licensed. To the extent that the metarule is a purported principle of Universal Grammar, this is a closure property that is exhibited by all GPSG grammars. Thus, the metarule is not so much a property defining a set of licensed structures as it is a property defining a class of sets of licensed structures—equivalently, a property of the definitions of such sets. If one is simply defining a particular set in that class it suffices to show that the set exhibits the property; there is no need to explicitly treat it as a characteristic of the class as a whole. Any interpretation that fails to do so, though, misses an important generalization that GPSG makes about the structure of language.

A similar account can be given of the GPSG claim that ID/LP grammars (in which the order of terms on the right hand side of the grammar rules is specified independently of the rules themselves) suffice for description of natural languages. The regularities such a format imposes on the set of trees so defined is, again, typically treated as an incidental property of that set. The specific logical languages employed in [BGMV93] and [BMV94], as the authors note, actually cannot express constraints in ID/LP format, although they could be modified to do so. Keller [Kel93], on the other hand, encodes ID and LP facts of the GPSG grammar independently and thus respects the claim, but there is nothing in his formalization that enforces it. Here again, is an important aspect of the GPSG theory of language, perhaps one of the aspects most subject to empirical verification, that is simply unaccounted for in the model-theoretic interpretations of it.

More problematic is the fact that there are aspects of GPSG, in particular the notion of Feature Specification Defaults, the distinction between features that are inherited and those that are instantiated, and the related notion of free features, that are so closely connected to the derivational conception of grammar that it is by no means obvious that they have a declarative interpretation—or at least not clear that they have one that does not explicitly encode a derivation. Consequently, it is not clear that it is even possible to capture these in an reasonable way within the model-theoretic approach.[5] Nonetheless, as with metarules and ID/LP format, these notions are the basis of generalizations that are important components of the GPSG theory of language.

Our model-theoretic reinterpretation explicitly incorporates those components of GPSG that, like metarules and ID/LP format, operate at the theoretic level. We demonstrate, then, that, at least for GPSG, it is possible to give a model-theoretic account of linguistic

sets of structures, might be considered a theoretic level—positing categories and constituency relationships—when considering languages as sets of strings. The distinction between weak and strong generative capacity, then, is analogous, in some sense, to the distinction between strong generative capacity and equivalence at the theoretic level. In fact, our argument that linguistic theories dwell partly at the theoretic level mimics arguments that they are concerned with strong, and not just weak, generation.

[5] As we have noted, these notions are given declarative definitions in GKP&S, but, since these are at least partly couched in terms of an algorithm for model checking, this fact is not obvious.

theories at both the descriptive and theoretic level, and to do so in a reasonably uniform manner.

1.2 Syntax from a Model-Theoretic Point of View

In the remainder of this paper we sketch some of the components of our model-theoretic reinterpretation of GPSG. We begin, in the next section, by exploring, in more detail, the nature of GPSG as a linguistic theory. We then look at how one can capture this theory within $L^2_{K,P}$, first, in Section 4, at the descriptive level and then, in Section 5, at the theoretical level. In the interest of brevity the discussion in Section 4 is abridged and we restrict our attention in Section 5 to the ECPO property. This provides a reasonably good glimpse of what is involved in capturing GPSG in $L^2_{K,P}$ and is sufficient foundation for a discussion of one of the issues for GPSG that this study uncovers—the theoretical difficulties partial categories present for ECPO—with which we close the paper. A complete discussion of the reinterpretation along with other issues it raises can be found in [Rog96b].

The model-theoretic approach we apply here to GPSG provides quite a powerful framework for studying theories of syntax. By stating the claimed properties of the structure of language directly as properties of a class of formal structures one gains considerable clarity in what exactly is being claimed as well as the ability to reason formally about the consequences of those claims. By working within restricted logical languages, like $L^2_{K,P}$, for which characterization results can be obtained, we can do this without losing the connection to the traditional notions of language theoretic complexity. Of course $L^2_{K,P}$ is relatively weak in language theoretic terms. The more powerful mechanisms underlying many contemporary linguistic theories are motivated, in part, by an assumption that the languages those theories license are not strongly context-free. Whether this is actually the case—whether the theories make use of the added power of the formalisms in a necessary way—is an open question, one that can be naturally approached from the model-theoretic point of view. We have used definability in $L^2_{K,P}$, for instance, to show that a reasonably standard GB account of English D-Structure and S-Structure actually licenses a strongly context-free language [Rog94, Rog96a]. On the other hand, in the same work we look at why the obvious GB analysis of the constructions discussed by Huybregts [Huy84] and Shieber [Shi85] are not amenable to definition within $L^2_{K,P}$. There is reason to believe, however, that the results characterizing the strongly context-free languages by definability in $L^2_{K,P}$ can be generalized to stronger languages and larger language theoretic complexity classes. We view this work as a model, then, for similar approaches, perhaps with stronger languages, to other linguistic formalisms.

2 The Role of the Grammar Formalism in GPSG

Perhaps the most fundamental characteristic of GPSG is that it is conceived in terms of the licensing of structures rather than their generation.[6] While the underlying backbone of a GPSG grammar has the form of Context-Free Grammar, albeit one specified in ID/LP

[6] This approach, of course, is not unique to GPSG. See, for instance, Peters and Ritchie's notion of Immediate Constituent Analysis [PR69], Joshi and Levy's Local Constraints [JL77, JLY80, JL82], or, outside the domain of trees, Johnson and Postal's Arc Pair Grammar [JP80].

format, these rules are not understood to specify a set of rewriting operations that together derive the strings in the intended language. Instead the notion of *derivation* is replaced with a notion of *admissibility*. The grammar rules are taken to specify a set of *local trees*—trees consisting of just a parent node and its set of children—and the grammar as a whole is taken to admit the set of all finite trees in which every local tree is drawn from that set.

Consequently, GPSG is a declarative formalism with much in common with the constraint-based formalisms that have largely superseded it; but despite having lost its procedural nature (and in contrast to the constraint-based formalisms) the grammar formalism in GPSG has a central role in expressing the linguistic theory. To quote GKP&S:

> The most interesting contribution that generative grammar can make to the search for universals of language is to specify formal systems that have putative universals as consequences, as opposed to merely providing a technical vocabulary of terms of which autonomously stipulated universals can be expressed. [GKPS85, pg. 2]

For GKP&S, the universal aspects of the theory should be expressed in the form of restrictions embodied in the grammar formalism. The linguistically germane generalizations are all and only those that can be expressed in the formalism; the consequences of the restrictions are expected to be characterize human languages. In this way the restricted nature of the formalism is understood to explain the regularities exhibited by the human languages as a class.

A basic example of the way in which properties of language are expressed in the restricted form of GPSG is the Exhaustive Constant Partial Ordering (ECPO) property. Stated in terms of phrase-structure rules, this holds iff the set of expansions of any one category observes a partial ordering with respect to precedence that is also observed by the expansions of all other categories. In more general terms, constraints on (i.e., regularities exhibited by) linear precedence are required to be independent of constraints on (i.e., regularities exhibited by) constituency. As a property of sets of trees, it requires that if there is any category A that labels nodes with children including both nodes labeled B and nodes labeled C for which the B invariably precedes the C, then it must be the case, for all trees in the set, that B precedes C whenever they label nodes in the same set of siblings.

The claim that the natural languages can be defined by sets of local trees that exhibit the ECPO property is a claim about the structure of those languages, a purported principle of Universal Grammar. In GPSG this claim is embodied by the ID/LP format of the rules. ID/LP grammars are incapable of specifying LP relationships between individual categories that are specific to the constituents of some particular category. This is the way in which, for GKP&S, "significant properties of grammars and languages fall out as theorems as opposed to being stipulated as axioms"[GKPS85, pg. 4]. In a sense, though, this is just moving the stipulation from the level of the grammar (where one might restrict oneself to sets of rules that exhibit the ECPO property) to that of the formalism (where one restricts oneself to a mechanism that can only express such sets of rules). The observed property is explained by the fact that it is a consequence of a claimed universal principle, but ID/LP format is not that principle, rather the ECPO property is. The principle justifies the restriction to ID/LP format, but short of claiming psychological reality for the formalism itself, one would not claim that the restriction to ID/LP format explains the ECPO property. In contrast, one might make a claim that human processing of precedence issues in syntax is effectively

independent of human processing of constituency issues. This serves as a justification of ECPO as a universal of language. The restriction of the formalism to ID/LP format is one way of stating that universal.

There are a couple of issues that arise when expressing language principles in grammar formalisms in this way. First, there is a question about the intended generality of the principles. In GPSG the ECPO property is expressed by the ID/LP format of the grammar. But GPSG actually consists of a number of modules in addition to the ID and LP rules: a system of Metarules expressing generalizations about the ID rules, Feature Co-occurrence Restrictions (FCRs) expressing generalizations about the features labeling individual nodes in the tree, principles governing the propagation of features within the tree (the Foot Feature Principle—FFP, Control Agreement Principle—CAP, and Head Feature Convention—HFC), and a system of defaults (Feature Specification Defaults—FSDs). The ID/LP rules constrain only the basic form of the local trees. A rule actually licenses any local tree in which the features attributed to the nodes extend the features in the rule, so long as they meet the additional principles embodied in FCRs, FSDs and the HFP, FFP, and CAP. But, given any set of local trees presented as a CFG, it is trivially possible to present it in ID/LP format if one can employ an extended set of features—if one can distinguish sub-categories on the basis of the licensing context.[7] It is not at all clear that such features cannot be introduced by these additional principles, converting a grammar which cannot be presented in ID/LP format into one that satisfies ECPO. More importantly, it is not clear what guarantees that these other components cannot introduce correlations between precedence relations and constituency relations, converting a grammar presented in ID/LP format into one for which ECPO fails. Consequently, requiring the grammar rules to be in ID/LP format does not appear to be, by itself, either necessary or sufficient to guarantee that the set of trees licensed by the grammar exhibits the ECPO property. Furthermore, it could be the case that violations of ECPO that are introduced by these other components of the grammar are in fact linguistically motivated exceptions. It is not clear whether the intended universal is that the ECPO property holds generally, or that it holds subject to such exceptions.

The second complication is the fact that the consequences of restrictions to a formalism generally extend beyond the principles they were intended to embody. To a large extent this is what is wanted. One wants a relatively compact set of universal principles to entail all of the regularities of human languages. So in proposing a restricted formalism one is making the claim that all of the consequences of the restrictions are in fact properties of human languages. The problem is that it is not immediately obvious that the consequences of the encoding of a principle in a grammar formalism necessarily coincide with the consequences of that principle. Asserting the ECPO property by restricting to ID/LP grammars, for instance, also restricts to languages that can be defined by sets of local trees, which implies a restriction to strongly context-free languages. This in turn implies certain closure properties for those languages, in particular Bar-Hillel's pumping lemma. Such a claim is falsifiable, and, if one accepts the evidence of Huybregts [Huy84] and Shieber [Shi85], amongst others, has been falsified. But the potential failure of the claim of strong context-freeness has no effect whatsoever on the claim for the ECPO property.

[7] This point is made in [GP82].

3 GPSG without Phrase-Structure Rules

In the following sections we explore these issues by looking at ECPO in the context of a model-theoretic reinterpretation of GPSG. The language we employ is $L^2_{K,P}$ [Rog94], a monadic second-order language over a signature including a set of individual constants ($\boldsymbol{K}$), a set of monadic predicates ($\boldsymbol{P}$), and binary predicates for domination ($\lhd^*$), immediate domination ($\lhd$), linear precedence ($\prec$), and equality ($\approx$). The syntax of this language allows us to make assertions about the relationship between pairs of nodes in terms of these binary relations, about membership relations between nodes and sets of nodes, and to quantify both over individual nodes and over arbitrary sets of nodes. Thus we can make assertions like:

$$(\forall X)[(\forall x, y)[(X(x) \wedge x \lhd^* y) \rightarrow X(y)] \rightarrow (\exists x)[X(x) \wedge \mathrm{P}(x)]]$$

which asserts that every set of nodes that is closed under domination (i.e., that contains every node dominated by any node in the set) includes some node labeled P.[8] We will continue to follow the conventions employed in this formula throughout the paper—set variables will be capitalized, individual variables will be in lower case, and membership of the individual assigned to x in the set assigned to X will be asserted as $X(x)$. As models, we take trees to be labeled subsets of $\mathcal{N}_\omega$—the complete ω-branching tree.[9] We can think of a labeled tree in this context as an assignment of labels ($\boldsymbol{P}$) to nodes in a subset of $\mathcal{N}_\omega$. Equivalently, we can interpret the labels as predicate variables, in which case the labeled tree defines an assignment of subsets of $\mathcal{N}_\omega$ to the labels. A tree satisfies a formula $\phi(\boldsymbol{P})$ over the labels in $\boldsymbol{P}$ iff the assignment encoded by that tree makes $\phi(\boldsymbol{P})$ true in $\mathcal{N}_\omega$. The theory of such structures is a definable fragment of SωS, the theory of multiple successor functions and is therefore decidable [Rab69].

There is a strong sense in which this logic is natural for reasoning about GPSG. The sets of finite trees with bounded branching that are definable in $L^2_{K,P}$ are exactly the *recognizable sets*—roughly, the sets of derivation trees generated by Context-Free Grammars. If we allow unbounded finite branching, the definable sets of trees are those generated by infinite CFGs that are themselves generated by Regular Grammars.[10] This generalizes slightly the class of sets of trees definable using the ID/LP format of GPSG in which the Kleene star can occur on the right hand side of ID rules.

Given that the class of trees definable in $L^2_{K,P}$ includes the class of trees that GKP&S argue is the class of trees definable in GPSG, we expect to be able to capture GPSG grammars on the descriptive level. The fact that we can do so confirms the claim of GKP&S that the strong generative capacity of the grammar is not extended beyond that of the ID/LP rules alone by the other components of the grammar. We are interested, however, with capturing not just the sets of trees licensed by GPSG grammars but also linguistic theory embodied in GPSG as well. It is here that $L^2_{K,P}$ and, in particular the interpretation of $L^2_{K,P}$ over subsets of $\mathcal{N}_\omega$, excels. Since we take models to be assignments of subsets to predicate variables, i.e., to be sequences of subsets, and since $L^2_{K,P}$ allows quantification over subsets, we can, in effect, quantify over models. This is exactly what is needed to

[8] There is no significance to this formula; it is intended only to demonstrate the syntax of $L^2_{K,P}$.

[9] This is the tree in which every node has an infinite set of children which, under the ordering of linear-precedence, is isomorphic to the natural numbers ordered by less-than.

[10] See Langendoen [Lan76] for some details on such grammars.

capture many of the principles of GPSG at the theoretic level. In this way we achieve a uniform formalization of both the descriptive level—formalized in GKP&S within the GPSG grammar—and the theoretic level—formalized in GKP&S at the metalevel. Note that because the formalization is still entirely within the language of $L^2_{K,P}$ every set of trees (i.e., every language) that is consistent with the theory it defines is strongly context-free (in the appropriately generalized sense), and the class of such sets of trees is decidable.

4 The Descriptive Level

In this section we will focus on GPSG at the descriptive level. Our goal is a purely declarative definition of the set of trees licensed by a given GPSG grammar (although, of course, we do not assume any given grammar in particular). We begin with the basic ontology of GPSG.

4.1 Labels and Categories

Following GKP&S, we assume we are given finite sets F and A of features and atomic feature-values, respectively, and that the set of all categories is constructed in stages from these. The set of all 0-level categories (K^0) is a subset of the set of all partial functions from F to A, where the subset is selected by two factors: the domain of the function is restricted to subsets of *Atom*, that subset of F that may take atomic feature-values, and the range of the function is restricted by an appropriateness function $\rho^0 : F \rightarrow \boldsymbol{\mathcal{P}}(A)$ which determines the subset of atomic features that each atomically valued feature may take. Both of these restrictions are determined by ρ^0, since $\rho^0(f)$ is non-empty iff $f \in Atom$.

Categories at level i (i.e., K^i) are all categories at level $i-1$ plus a set of functions from F to K^{i-1}, similarly constrained by ρ^i and a requirement restricting the value of a category for a feature f to categories that are not themselves defined for f at any level. Again, both constraints can be determined by ρ^i.

The set of all categories is K^n, where n is the cardinality of $F \setminus Atom$.

To capture this, we take our basic set of labels to be sequences in F^*A, in which no feature repeats. Each sequence of length $i > 1$ represents a path through a category of level $i-2$. We extend this basic set of labels in three ways. First we will include prefixes of these paths, i.e., strings in F^*. This will allow us to reason about the set of all nodes which are assigned to categories defined for a given feature regardless of the value for which that feature is defined. There is an implicit Feature Co-occurrence Constraint that requires a node labeled $P \in (F^*) \cup (F^*A)$ to be also labeled P' for all $P' \in F^*$ that are prefixes of P. Second, we will assume certain additional labels picking out linguistically significant subsets of nodes (in particular H which labels nodes that are heads). We will ignore these for now and take $\boldsymbol{P}$ to be just the set of sequences in $(F^*) \cup (F^*A)$ that correspond to (prefixes of) paths that actually occur in categories in K^n. We will refer to the sequences in $F^*A \cap \boldsymbol{P}$ as $\mathbf{P}^T$. Finally, we will augment the labels in $\boldsymbol{P}$ (i.e., those that actually occur in the signature of our models) with certain defined labels (monadic predicates). Most notable among these is the set of categories $\boldsymbol{C}$.[11]

[11] We use $\boldsymbol{C}$ rather than $\boldsymbol{K}$ which we reserve for the set of individual constants in $L^2_{K,P}$, although such constants play no role here.

Note that, in our reconstruction, categories, as defined in GKP&S, correspond to sets of atomic formulae over $\mathbf{P}^T$—each category corresponds to the set of paths for which it is defined. Let

$$\boldsymbol{C} = \left\{ C \subseteq \mathbf{P}^T \mid C \text{ is the set of paths occurring in a category in } K^n \right\}.$$

For each $C \in \boldsymbol{C}$ we define a predicate picking out those nodes labeled with (an extension of) the corresponding category:[12]

$$C(x) \equiv \bigwedge_{P \in C} [P(x)].$$

This is the notion of category that is employed in GKP&S, where partial categories are given full status as categories in there own right. So [+V] picks out the category generalizing verbs and prepositions, and every node assigned to category [−N, +V] or [+N, +V] is also assigned to [+V]. There are, however, occasions in which we need to reason in terms of the actual instances of these categories occurring in the trees, that is, in terms of the exact set of attributes assigned to a node (as distinct from the set of all categories extending those attributes). For these cases we define for each $C \in \boldsymbol{C}$, a predicate $C^T(x)$:

$$C^T(x) \equiv \bigwedge_{P \in C} [P(x)] \wedge \bigwedge_{\bar{P} \in \mathbf{P}^T \setminus C} [\neg \bar{P}(x)].$$

We will refer to the set of all such predicates as $\boldsymbol{C}^T$.

It is frequently useful to consider these predicates in terms of the sets of nodes that they label in some tree, i.e., $C \in \boldsymbol{C}$ can be regarded as (assuming all trees have distinct universes)

$$\{a \mid \text{for some tree } T,\ a \in |T| \text{ and } T \models C(a)\}.$$

So each $C(x)$ picks out the set of nodes assigned to category C, each $C^T(x)$ picks out the set of nodes assigned to category C and none of its extensions, and each $P \in \boldsymbol{P}$ picks out the set of nodes assigned to a category defined for P.

Abbreviations

We will follow the usual linguistic conventions in abbreviating features. For instance [ACC](x) will be understood as $\langle \text{CASE}, \text{ACC} \rangle (x)$. Similarly

$$\text{NP}(x) \equiv \text{N}^2(x) \equiv \langle \text{N}, + \rangle (x) \wedge \langle \text{V}, - \rangle (x) \wedge \langle \text{BAR}, 2 \rangle (x)$$

and

$$\text{NP[ACC]}(x) \equiv \text{NP}(x) \wedge \text{[ACC]}(x).$$

[12] There are several obvious ways to extend this. First, we might take categories to be sets of atomic formulae over $\boldsymbol{P}$. This allows categories to be specified as being defined for a feature without being specified as being defined for any particular value for that feature. The categories of [GP82], in fact, correspond to just such sets of formulae. Second, we might take categories to be sets of literals rather than atomic formulae. This allows categories to be specified as being not defined for certain features (or, combining both these extensions, undefined for certain features). Finally, we might allow categories to correspond to formulae more complicated than conjunctions of sets of literals. This gets into the realm of feature-structure description logics, albeit those without re-entrancy.

Note also that, in GPSG convention, S and VP are taken to be

$$\begin{aligned} \mathrm{S}(x) &\equiv \mathrm{V}^2[+\mathrm{SUBJ}](x) \\ \mathrm{VP}(x) &\equiv \mathrm{V}^2[-\mathrm{SUBJ}](x). \end{aligned}$$

It doesn't matter whether we take these to be abbreviations at the meta-level, or we take them to be explicitly defined predicates.

4.2 ID Rules

ID rules license local trees. We begin with a family of predicates that identify local trees in which the children of x are exactly the distinct nodes in the sequence $\vec{y}$.

$$\mathrm{Children}(x, \vec{y}) \equiv \bigwedge_{y_i \in \vec{y}} [x \lhd y_i] \wedge \bigwedge_{i \neq j} [y_i \not\approx y_j] \wedge (\forall z)[x \lhd z \rightarrow \bigvee_{y_i \in \vec{y}} [z \approx y_i]].$$

With this, we can interpret each ID rule as a formula in which x and $\vec{y}$ occur free which is made true exactly at sets of nodes satisfying the rule—at sets of nodes that form local trees in which the nodes are labeled with extensions of the categories in the rule. In the terminology of GKP&S, it is made true by local trees that are *induced* by the rule. For the ordinary ID rules this is completely straightforward.

$$\begin{aligned} & \mathrm{VP} \longrightarrow \mathrm{H}[5],\ \mathrm{NP},\ \mathrm{NP} \\ \mathrm{ID}_5(x, y_1, y_2, y_3) \equiv\ & \mathrm{Children}(x, y_1, y_2, y_3) \wedge \mathrm{VP}(x) \wedge \\ & \mathrm{H}(y_1) \wedge \langle \mathrm{SUBCAT}, 5 \rangle\,(y_1) \wedge \mathrm{NP}(y_2) \wedge \mathrm{NP}(y_3). \end{aligned}$$

The iterating coordination schema (CS^+) is the only infinite set of ID rules in GKP&S. This we treat as a finite set of "greater-than-1" branching rules

$$\begin{aligned} & \mathrm{X} \longrightarrow \mathrm{H}[\mathrm{CONJ}_{\alpha_0}],\ \mathrm{H}[\mathrm{CONJ}_{\alpha_1}]^+ \\ \mathrm{CS}^+(x, y_1) \equiv\ & \mathrm{H}(y_1) \wedge \langle \mathrm{CONJ}, \alpha_0 \rangle\,(y_1) \wedge (\exists z)[x \lhd z \wedge z \not\approx y_1] \wedge \\ & (\forall z)[x \lhd z \rightarrow (z \approx y_1 \vee \mathrm{H}(z) \wedge \langle \mathrm{CONJ}, \alpha_1 \rangle\,(z))]. \end{aligned}$$

Other schemata with unbounded right-hand-sides can be treated similarly.

At this level, we take metarules simply to be abbreviatory conventions. As we noted above, these are principles of the theoretic level of GPSG, amenable to a treatment much like the treatment of ECPO in the next section.[13] At the descriptive level, metarules just specify finite sets of additional ID rules, formed from the basic rules explicitly presented in the grammar. We will assume that our set of ID formulae is formed from the ID rules following expansion of the metarules.

We will refer to these formulae translating the ID rules as the *local contexts*[14] of the grammar. Let $\mathbf{ID}_G$ be the set of all local contexts for the grammar G. Since the infinite

[13] See [Rog96b] for details.

[14] We distinguish local contexts from local trees. A local tree is labeled with some specific set of features and no others; we think of these in terms of $\mathbf{C}^T$, the fully defined categories. A local context, in contrast, picks out some set of local trees—those that satisfy the formula encoding the context.

schemata are translated as single rules, this set is finite. Since the translations of the infinite schemata have boundedly many free variables (x and y_1 here) the number of distinct free variables occurring free in the formulae of $\mathbf{ID}_G$ is bounded. Let $m+1$ equal this bound. (In the grammar of GKP&S $m = 3$ before application of the metarules, 4, or perhaps 5, after.) Tree admissibility, then, for the ID rules alone simply requires that each set of $m+1$ (not necessarily distinct) nodes in which one is the parent of each of the others satisfies one of the local contexts of $\mathbf{ID}_G$:

$$(\forall x, y_1, \ldots, y_m)[\bigwedge_{i \leq m} [x \lhd y_i] \rightarrow \bigvee_{\mathrm{ID} \in \mathbf{ID}_G} [\mathrm{ID}(x, y_1, \ldots, y_m)]].$$

4.3 LP Rules and FCRs

FCRs translate directly:

$$[+\mathrm{INV}] \supset [+\mathrm{AUX},\ \mathrm{FIN}]$$

becomes

$$(\forall x)[[+\mathrm{INV}](x) \rightarrow [+\mathrm{AUX},\ \mathrm{FIN}](x)].$$

Linear Precedence rules are slightly more complicated in that they are stated in terms of "not-right-of" or, equivalently, "left-of-or-equal-to" (this is the relation $\prec$ in GKP&S) and they apply only to sets of siblings. So we translate

$$[+\mathrm{N}] \prec \mathrm{P}^2$$

as

$$(\forall x, y)[\mathrm{Siblings}(x, y) \rightarrow ([+\mathrm{N}](x) \wedge \mathrm{P}^2(y) \rightarrow y \not\lhd x)],$$

where

$$\mathrm{Siblings}(x, y) \equiv (\exists z)[z \lhd x \wedge z \lhd y].$$

Similarly,

$$[\mathrm{SUBCAT}] \prec\!\sim [\mathrm{SUBCAT}]$$

translates as

$$(\forall x, y)[\mathrm{Siblings}(x, y) \rightarrow ([\mathrm{SUBCAT}](x) \wedge \neg[\mathrm{SUBCAT}](y) \rightarrow y \not\lhd x)].$$

4.4 Agreement Principles

Issues like projection of head features, filler/gap dependencies, and other agreement issues are handled, in GPSG, by a system of principles which require pairs of nodes occurring in particular configurations in the same local tree to agree on certain classes of features. The effect of these principles is the propagation of features between pairs of nodes standing in agreement relationships like projection/head, filler/gap, subject/verb, etc. There are three of these principles: the Head Feature Convention, which governs projection issues; the Foot Feature Principle, which covers issues like filler/gap dependencies; and the Control Agreement Principle, which covers traditional agreement issues. For the most part, encoding the relevant configurations in $L^2_{K,P}$ is easy, as is encoding the requirement that specific sets of features co-occur in pairs satisfying those configurations. What complicates theses principles

is the way in which they are overridden by the ID rules and FCRs, and the way in which they override each other.

For the FFP, exceptions are just the *inherited features*—features that are stipulated by the ID rules (including those instantiating metarules). We can define the property of a node bearing an inherited feature using the encoding of the ID rules. For example if ID_{42} and ID_{44} are ID rules stipulating the feature f on the second and third child respectively, we can identify nodes inheriting f with:[15]

$$\begin{aligned}\mathrm{Inh}_f(x) \quad \equiv \quad & (\exists y_0, y_1, y_2)[\mathrm{ID}_{42}(y_0, y_1, y_2) \wedge x \approx y_2] \vee \\ & (\exists y_0, y_1, y_2, y_3)[\mathrm{ID}_{44}(y_0, y_1, y_2, y_3) \wedge x \approx y_3] \vee \\ & \vdots\end{aligned}$$

With such formulae we can explicitly incorporate exceptions into the definition of the FFP. The case of CAP is considerably more complicated, but can be treated in essentially the same way.[16] The HFC is overridden not only by the ID rules, but also by FCRs, the FFP and the CAP. Here, GKP&S define a notion of a node being *free* to take a feature—of that feature being compatible with the features assigned to a node by the ID rules, the FCRs, the FFP and CAP. Again, we can use the formulae encoding these other principles to define a predicate $\mathrm{Free}_f(x)$ that is made true at a node only if it is free to take feature f. With this we can capture the HFC.

4.5 Feature Specification Defaults

The final class of universal feature instantiation principles are the Feature Specification Defaults (FSDs). FSDs specify conditions on feature values that must hold at a node in a licensed tree unless they are overridden by some other component of the grammar; in particular, unless they are incompatible with either an inherited feature or a feature required by one of the agreement principles. It is the fact that the default holds just in case it is incompatible with these other components that gives FSDs their dynamic flavor. Note, though, FSDs are a static sort of default. A GPSG grammar is not an evolving theory; the exceptions to the defaults are fully determined when the grammar is written. If we ignore for the moment the effect of the agreement principles, the defaults are roughly the converse of the ID rules: a non-default feature occurs iff it is licensed by an ID rule. This situation can be captured in $L^2_{K,P}$ using the encoding of $\mathrm{Free}_f(x)$ sketched above.

Since the agreement principles require pairs of nodes occurring in certain configurations in local trees to agree on certain classes of features, they do not introduce features into the trees, but rather propagate features from one node to another, possibly in many steps. Consequently, these principles cannot override FSDs by themselves; rather every violation of a default must be licensed by an inherited feature somewhere in the tree. In order to account for this propagation of features, the definition of FSDs in GKP&S is based on identifying pairs of nodes that co-vary wrt the relevant features in all possible extensions of the given

[15] Note that it may be necessary to explicitly distinguish contexts which satisfy multiple ID rules, some of which stipulate a feature and others of which do not. In the extreme we could label the root, say, of each local tree with the name of its licensing rule.

[16] For CAP, inherited features are not exceptions, but rather certain features become control features by virtue of being inherited.

tree. As a result, although the treatment in GKP&S is actually declarative, this fact is far from obvious.

Again, it is not difficult to define the configurations of local trees in which nodes are required to agree by FFP, CAP, or HFC in $L^2_{K,P}$. Let the predicate $\text{Propagate}_f(x, y)$ hold for a pair of nodes x and y iff they are required to agree on f by one of these principles (and are, thus, in the same local tree). Note that Propagate is symmetric. Following the terminology of GKP&S, we can identify the set of nodes that are prohibited from taking feature f by the combination of the ID rules, FFP, CAP, and HFC as the set of nodes that are *privileged* wrt f. This includes all nodes that are not Free for f as well as any node connected to such a node by a sequence of Propagate_f links. We, in essence, define this inductively. $\text{P}'_f(X)$ is true of a set iff it includes all nodes not Free for f and is closed wrt Propagate_f. $\text{PrivSet}_f(X)$ is true of the smallest such set.

$$\begin{aligned}
&\text{P}'_f(X) \equiv \\
&\quad (\forall x)[\neg\text{Free}_f(x) \rightarrow X(x)] \wedge \\
&\quad (\forall x)[(\exists y)[X(y) \wedge \text{Propagate}_f(x, y)] \rightarrow X(x)]
\end{aligned}$$

$$\begin{aligned}
\text{PrivSet}_f(X) \quad \equiv \quad & \text{P}'_f(X) \wedge \\
& (\forall Y)[\text{P}'_f(Y) \rightarrow \text{Subset}(X, Y)].
\end{aligned}$$

There are two things to note about this definition. First, in any tree there is a unique set satisfying $\text{PrivSet}_f(X)$ and this contains exactly those nodes not Free for f or connected to such a node by Propagate_f. Second, while this is a first-order inductive property, the definition is a second-order explicit definition. In fact, the second-order quantification of $L^2_{K,P}$ allows us to capture any monadic first-order inductively or implicitly definable property explicitly.

Armed with this definition, we can identify individuals that are privileged wrt f simply as the members of PrivSet_f.[17]

$$\text{Privileged}_f(x) \equiv (\exists X)[\text{PrivSet}_f(X) \wedge X(x)].$$

One can define $\text{Privileged}_{\neg f}(x)$ which holds whenever x is required to take the feature f along similar lines.

These, then, let us capture FSDs. For the default [−INV], for instance, we get:

$$(\forall x)[\neg\text{Privileged}_{[-\,\text{INV}]}(x) \rightarrow [-\,\text{INV}](x)].$$

For [BAR 0] $\supset\sim$ [PAS] (which says that [Bar 0] nodes are, by default, not marked passive), we get:

$$\begin{aligned}
(\forall x)[\ & ([\text{BAR}\,0](x) \wedge \neg\text{Privileged}_{\neg[\text{PAS}]}(x)) \rightarrow \\
& \neg[\text{PAS}](x)].
\end{aligned}$$

The key thing to note about this treatment of FSDs is its simplicity relative to the treatment of GKP&S. The second-order quantification allows us to reason directly in terms of the sequence of nodes extending from the privileged node to the local tree that actually licenses the privilege. The immediate benefit is the fact that it is clear that the property of satisfying a set of FSDs is a static property of labeled trees and does not depend on the particular strategy employed in checking the tree for compliance.

[17] We could, of course, skip the definition of PrivSet_f and define $\text{Privileged}_f(x)$ as $(\forall X)[\text{P}'(X) \rightarrow X(x)]$, but we prefer to emphasize the inductive nature of the definition.

4.6 Tree Admissibility

Given any GPSG grammar G, using the conjunction of the formulae sketched in this section we can construct a formula $\Phi_G(\mathbf{P})$ over the labels in $\boldsymbol{P}$[18] which will be satisfied by a tree labeled with $\boldsymbol{P}$ iff that tree is licensed by G. These formulae, then, give us a definition in the language $L^2_{K,P}$ of the set of trees licensed by G. One consequence of this is a confirmation that the additional mechanisms of GPSG (metarules, FCRs, FFP, CAP, HFC, and FSDs) do not extend the generative capacity of the formalism beyond that of the underlying phrase structure grammar, i.e., beyond that of CFGs which are themselves generated by Regular Grammars.

5 The Theoretic Level

The previous section sketches how we can capture a given GPSG grammar at the descriptive level in $L^2_{K,P}$. Thus there is a class of $L^2_{K,P}$ formulae that define GPSG grammars. In this section we explore how to capture GPSG at the theoretic level within $L^2_{K,P}$. Whereas in GKP&S the principles embodying the linguistic theory are expressed as restrictions to the class of grammars, here we will be expressing those principles as properties of the sets of models defined at the descriptive level.[19] Together these properties characterize the sets of trees generated by GPSG grammars. We can abstract away from the form of the definition—from Φ_G—and say that any $L^2_{K,P}$ formula that defines a set of trees that satisfies these properties is a presentation of a grammar within the linguistic theory expressed by GPSG.[20] Effectively we translate GKP&S's restricted class of grammars to a restricted class of definitions of sets of trees.

5.1 The ECPO Property

Because Φ_G is a translation of a GPSG grammar, ID clauses and LP clauses will be independent. But we don't want to claim that the ID/LP format is somehow characteristic of natural language, rather we want to capture the assertion that all human languages exhibit the ECPO property. Instead of restricting our attention to definitions in $L^2_{K,P}$ that have an ID/LP format (like Φ_G) we will restrict ourselves to definitions of sets of trees that exhibit the ECPO property—to those that are (roughly) logically equivalent to definitions in which ID and LP constraints are stated independently.

The ECPO property is a closure property of sets of trees. Ideally, we would like a direct $L^2_{K,P}$ statement of this property. While we can't quite do that, we can, given an arbitrary $L^2_{K,P}$ definition Φ_G, express the proposition that the set of trees that satisfy Φ_G exhibits ECPO.

[18] Note that a more consistent notation would be $\Phi_G(\vec{\mathrm{P}})$ explicitly treating $\boldsymbol{P}$ as a sequence of monadic predicates (or, perhaps, free set variables), but we use the form given in order to clarify where they come from.

[19] The fact that we restrict ourselves to $L^2_{K,P}$ already establishes one of the most basic properties of these sets—that they are strongly Context-Free (in a suitably generalized sense).

[20] We do not, on the other hand, want to argue that every such formula is equivalent to a GPSG grammar, since the encoding of the theory in the GPSG formalism may entail more than just the linguistic theory it is intended to capture.

We begin by defining an auxiliary predicate. Let

$$\tau_{2,3}(x,y,z) \equiv x \lhd y \wedge x \lhd z \wedge C_2^T(y) \wedge C_3^T(z)$$

where C_2^T and C_3^T are categories in $\boldsymbol{C}^T$. This holds at a triple of points iff the first is the parent of the other two and those other two are labeled with exactly the categories C_2 and C_3, respectively.

Then the ECPO property (with respect to $\boldsymbol{C}^T$) is:

ECPO:

$$\bigwedge_{C_2^T, C_3^T \in \boldsymbol{C}^T} \Big[\quad \bigvee_{C_1^T \in \boldsymbol{C}^T} \Big[\quad (\exists \boldsymbol{P})\big[\Phi_G(\boldsymbol{P}) \wedge (\exists x,y,z)[C_1^T(x) \wedge \tau_{2,3}(x,y,z)]\big] \wedge$$
$$(\forall \boldsymbol{P})\big[\Phi_G(\boldsymbol{P}) \rightarrow (\forall x,y,z)[C_1^T(x) \wedge \tau_{2,3}(x,y,z) \rightarrow z \not\lhd y]\big]\Big] \rightarrow$$
$$(\forall \boldsymbol{P})\big[\Phi_G(\boldsymbol{P}) \rightarrow (\forall x,y,z)[\tau_{2,3}(x,y,z) \rightarrow z \not\lhd y]\big] \qquad \Big]$$

In words, **ECPO** states that: for all categories C_2 and C_3, if there is any category C_1 that exactly labels the parent of nodes labeled exactly C_2 and C_3 in some tree satisfying Φ_G for which, in every such case, the child labeled C_3 does not precede that labeled C_2, then it is universally true that nodes labeled exactly C_3 will never precede those labeled exactly C_2 whenever they occur as siblings in any tree that satisfies Φ_G.

There are a number of things to note about **ECPO**. First, the big wedge and vee are, of course, schematic representations of conjunctions and disjunctions. In particular, they are finite (since $\boldsymbol{C}_T$ is) and they do not bind variables. None of the C_i^T occur outside the scope of the quantifiers in this formula. Secondly, we are treating the predicates in $\boldsymbol{P}$ as set variables. Each of their occurrences in the C_i^T are bound by the quantifiers. Note also, that the formula is a schema—given any $L^2_{K,P}$ formula Φ_G, we can plug it into **ECPO** to produce a sentence that is true of $\mathcal{N}_\omega$ iff the set of trees that satisfy Φ_G exhibits the ECPO property. Φ_G, in essence, "occurs free"—**ECPO** is a statement of a closure property on sets of models which is stated as a property of definitions. So while GPSG encodes the ECPO property by restricting the class of grammars, here we encode it by restricting the class of definitions. In contrast to the GPSG encoding of ECPO, we restrict the consequences of the definition, not its form. Every $L^2_{K,P}$ formula is a candidate for Φ_G. Note finally, that, as Φ_G is an $L^2_{K,P}$ formula, **ECPO** is also an $L^2_{K,P}$ formula. Thus, given some Φ_G, it is in principle possible to determine if **ECPO** holds. This is a decidable class of definitions.

We have based the definition of ECPO on the $\boldsymbol{C}^T$ notion of category rather than the $\boldsymbol{C}$ notion. This is a reasonable interpretation of ECPO as a property of sets of trees. What it says is that if we consider the labels that actually occur on the nodes in trees in the set, then there will be no correlation between the order in which those labels are distributed among siblings and the label of their parent. We might, on the other hand, base the definition on the $\boldsymbol{C}$ notion of category. This is, after all, GKP&S's notion of category. But, it turns out that ECPO defined in this way is too strong.

Consider, for instance, the following grammar fragment:

$$\begin{array}{rcl} \mathrm{S} & \longrightarrow & \mathrm{NP\,VP} \\ \mathrm{VP} & \longrightarrow & \mathrm{V\,VP} \\ \mathrm{VP} & \longrightarrow & \mathrm{V\,NP} \\ \mathrm{VP} & \longrightarrow & \mathrm{V\,NP\,VP}. \end{array}$$

This is just a CFG, it is not in ID/LP format, but the sets of trees it licenses do satisfy ECPO if we consider the set of categories $\{\mathrm{S}, \mathrm{NP}, \mathrm{VP}, \mathrm{V}\}$. Expanding VP, etc., to their equivalent sets of features we get something like:

$$\begin{array}{rcl} \mathrm{S} & = & [-\mathrm{N},\, +\mathrm{V},\, +\mathrm{SUBJ},\, \mathrm{BAR2}] \\ \mathrm{VP} & = & [-\mathrm{N},\, +\mathrm{V},\, -\mathrm{SUBJ},\, \mathrm{BAR2}] \\ \mathrm{NP} & = & [+\mathrm{N},\, -\mathrm{V},\, \mathrm{BAR2}] \\ \mathrm{V} & = & [-\mathrm{N},\, +\mathrm{V},\, \mathrm{BAR0}] \end{array}$$

and the grammar becomes:

$$\begin{array}{l} [-\mathrm{N}, +\mathrm{V}, +\mathrm{SUBJ}, \mathrm{BAR2}] \quad \longrightarrow \\ \qquad\qquad [+\mathrm{N}, -\mathrm{V}, \mathrm{BAR2}]\ [-\mathrm{N}, +\mathrm{V}, -\mathrm{SUBJ}, \mathrm{BAR2}] \\ [-\mathrm{N}, +\mathrm{V}, -\mathrm{SUBJ}, \mathrm{BAR2}] \quad \longrightarrow \quad [-\mathrm{N}, +\mathrm{V}, \mathrm{BAR0}]\ [+\mathrm{N}, -\mathrm{V}, \mathrm{BAR2}] \\ [-\mathrm{N}, +\mathrm{V}, -\mathrm{SUBJ}, \mathrm{BAR2}] \quad \longrightarrow \\ \qquad [-\mathrm{N}, +\mathrm{V}, \mathrm{BAR0}]\ [+\mathrm{N}, -\mathrm{V}, \mathrm{BAR2}]\ [-\mathrm{N}, +\mathrm{V}, -\mathrm{SUBJ}, \mathrm{BAR2}]\ . \end{array}$$

Here again, the set of trees this licenses satisfies ECPO over the categories that actually label nodes in the trees. But if we consider $\boldsymbol{C}$—the set of all partial categories—we will need to account for the categories [+SUBJ], [−SUBJ], [+N, −V], and [−N, +V] as well, and here we find that [+N, −V] always precedes [−N, +V] in expansions of [+SUBJ], but can occur in either order in expansions of [−SUBJ].

Clearly GKP&S intend ECPO to be interpreted with respect to some appropriate set of categories. But it is not clear which set of categories that is. One obvious choice is the one that we make here—require ECPO to hold over the set of all categories that actually label nodes that occur in the set of licensed trees. Another obvious choice would be to restrict it to the set of all categories occurring in the ID and LP rules, or perhaps to all extensions of those categories. The point is that the choice needs to be made explicit, and it probably ought to be linguistically motivated.

6 Conclusions

It is possible to develop useful theories of syntax purely within the model-theoretic approach we have demonstrated here, and with descriptive complexity results like the characterization of strongly context-free sets by definability in $L^2_{K,P}$ it is even possible to do this without losing traditional measures like language theoretic and computational complexity. For the most part, though, work within model-theoretic approach has focused on re-interpreting existing theories rather than addressing the syntax of languages directly,[21] and there may

[21] The notable exception is Johnson and Postal's Arc-Pair Grammar [JP80].

be good reasons for maintaining this relationship. The close connection between grammar-based theories and their underlying mechanism is a distinct advantage, for instance, when one undertakes to actually implement a parser based on the theory. We believe, though, that the model-theoretic approach is useful even if it serves only as an adjunct to grammar-based approaches, not only because of the way in which it can clarify the consequences of the formalism, but also because it has the potential to provide a uniform representation for a variety of formalisms. One might, in fact, hope to reduce distinctions between the formalisms underlying linguistic theories to distinctions between the properties of the sets of structures they license, perhaps illuminating the empirical consequences of those distinctions. If we are going to do this, however, we will need to broaden the focus of the model-theoretic approach beyond the theories of individual languages to encompass theories of natural languages in general.

References

[BGMV93] Patrick Blackburn, Claire Gardent, and Wilfried Meyer-Viol. Talking about trees. In *EACL 93*, pages 21–29. European Association for Computational Linguistics, 1993.

[BMV94] Patrick Blackburn and Wilfried Meyer-Viol. Linguistics, logic, and finite trees. *Bulletin of the IGPL*, 2(1):3–29, March 1994.

[Bre82] Joan Bresnan, ed. *The Mental Representation of Grammatical Relations*. MIT Press, 1982.

[Car92] Bob Carpenter. *The Logic of Typed Feature Structures; with Applications to Unification Grammars, Logic Programs and Constraint Resolution*. Number 32 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1992.

[Cho86] Noam Chomsky. *Knowledge of Language: Its Nature, Origin and Use*. Praeger, New York, 1986.

[Cor92] Thomas Longacre Cornell. *Description Theory, Licensing Theory, and Principle-Based Grammars and Parsers*. PhD thesis, University of California Los Angeles, 1992.

[DVS90] Anuj Dawar and K. Vijay-Shanker. An interpretation of negation in feature structure descriptions. *Computational Linguistics*, 16(1):11–21, 1990.

[GKPS85] Gerald Gazdar, Ewan Klein, Geoffrey Pullum, and Ivan Sag. *Generalized Phrase Structure Grammar*. Harvard University Press, 1985.

[GP82] Gerald Gazdar and Geoffrey K. Pullum. *Generalized Phrase Structure Grammar: A Theoretical Synopsis*. Indiana University Linguistics Club, 1982.

[GPC+88] Gerald Gazdar, Geoffrey Pullum, Robert Carpenter, Ewan Klein, T. E. Hukari, and R. D. Levine. Category structures. *Computational Linguistics*, 14:1–19, 1988.

[Huy84] Riny Huybregts. The weak inadequacy of context-free phrase structure grammars. In Ger J. de Haan, Mieke Trommelen, and Wim Zonneveld, editors, *Van Periferie Naar Kern*, pages 81–99. Foris Publications, Dordrecht, 1984.

[JL77] Aravind K. Joshi and Leon S. Levy. Constraints on structural descriptions: Local transformations. *SIAM Journal on Computing*, 6(2):272–284, June 1977.

[JL82] Aravind K. Joshi and Leon S. Levy. Phrase structure trees bear more fruit than you would have thought. *American Journal of Computational Linguistics*, 8(1):1–11, 1982.

[JLY80] Aravind K. Joshi, Leon S. Levy, and Kang Yueh. Local constraints in programming languages part I: Syntax. *Theoretical Computer Science*, 12:265–290, 1980.

[Joh88] Mark Johnson. *Attribute-Value Logic and the Theory of Grammar*. Number 16 in CSLI Lecture Notes. Center for the Study of Language and Information, Stanford, CA, 1988.

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

[JP80] David E. Johnson and Paul M. Postal. *Arc Pair Grammar*. Princeton University Press, Princeton, New Jersey, 1980.

[Kel93] Bill Keller. *Feature Logics, Infinitary Descriptions and Grammar*. Number 44 in CSLI Lecture Notes. Center for the Study of Language and Information, 1993.

[KR86] Robert T. Kasper and William C. Rounds. A logical semantics for feature structures. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, 1986.

[KR90] Robert T. Kasper and William C. Rounds. The logic of unification in grammar. *Linguistics and Philosophy*, 13:35–58, 1990.

[Kra95] Marcus Kracht. Syntactic codes and grammar refinement. *Journal of Logic, Language, and Information*, 4:41–60, 1995.

[Lan76] D. Terrence Langendoen. On the weak generative capacity of infinite grammars. *CUNYForum*, 1:13–24, 1976.

[MR87] M. Drew Moshier and William C. Rounds. A logic for partially specified data structures. In *ACM Symposium on the Principles of Programming Languages*, 1987.

[PR69] P. Stanley Peters and Robert W. Ritchie. Context-sensitive immediate constituent analysis—context-free languages revisited. In *Proceedings of the ACM Conference on the Theory of Computing*, pages 1–8. ACM, May 1969.

[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.

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

[Rog96a] James Rogers. On descriptive complexity, language complexity, and GB. In Patrick Blackburn and Maarten de Rijke, editors, *Specifying Syntactic Structures*. CSLI, 1996. Also available as IRCS Technical Report 95-14. cmp-lg/9505041.

[Rog96b] James Rogers. What does a grammar formalism say about a language? Technical Report 96-10, Institute for Research in Cognitive Science, University of Pennsylvania, 1996.

[RVS94] James Rogers and K. Vijay-Shanker. Obtaining trees from their descriptions: An application to tree-adjoining grammars. *Computational Intelligence*, 10:401–421, 1994.

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

[Smo89] Gert Smolka. A feature logic with subsorts. LILOG Report 33, IBM Germany, Stuttgart, 1989.

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