Bericht Nr. 72, Februar 1996

Arbeitspapiere des
Sonderforschungsbereichs 340

ISSN 0947-6954/96

Sprachtheoretische Grundlagen für die Computerlinguistik

James Rogers :

Capturing Linguistic Theories Model-Theoretically

Universität Stuttgart Universität Tübingen
IBM Deutschland Informationssysteme GmbH

Arbeitspapiere des SFB 340
Bericht Nr. 72, Februar 1996

CAPTURING LINGUISTIC THEORIES MODEL-THEORETICALLY

JAMES ROGERS

Abstract. Over the last ten or fifteen years there has been a shift, in generative linguistics, away from formalisms based on a procedural interpretation of grammars towards constraint-based formalisms—formalisms that define languages by specifying a set of constraints that characterize the set of well-formed structures analyzing the strings in the language. A natural extension of this trend is to define this set of structures model-theoretically—to define it as the set of mathematical structures that satisfy some set of logical axioms. In this paper we sketch $L^2_{K,P}$, a monadic second-order framework for axiomatizing theories of syntax. We look, in particular at two components of Generalized Phrase-Structure Grammar (GPSG)—Feature Specification Defaults (FSDs) and the Exhaustive Constant Partial Ordering property (ECPO)—that illustrate the strengths of $L^2_{K,P}$ as a framework.

FSDs have been cited as a component of GPSG which cannot be captured declaratively. We show that the apparent dynamic aspect of these defaults is a consequence of the difficulty of capturing them within the formal framework in which GPSG was originally defined. The monadic second-order quantification of $L^2_{K,P}$ allows FSDs to be defined in a much more direct way, clarifying their nature significantly.

ECPO, despite being a fundamental principle of GPSG, has been ignored by existing model-theoretic treatments of GPSG. We argue that this is a consequence of the fact that these treatments focus on axiomatizing the set of trees licensed by a given GPSG grammar, while ECPO, being a universal principle of natural language, is a characteristic of the class of sets of trees that analyze natural languages. That is, it is not a property of trees, but is, rather, a property of sets of trees. $L^2_{K,P}$, again principally because it supports second-order quantification, allows us to capture, in a limited way, universal principles like ECPO.

Finally, we note that, while $L^2_{K,P}$ is powerful enough to capture principles like FSDs and ECPO in a natural way, its formal expressive power is well-defined. The class of finite trees that are definable in $L^2_{K,P}$ are a very slight generalization of the class of sets of trees definable in the phrase-structure formalism underlying GPSG.

This is a preliminary version of a more complete treatment of this work that is in preparation [Rog].

ISSN 0947-6954/96

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. We won't dwell on what motivated this concept of grammar. Presumably, it can be traced to an intent to model the mechanism of the human language faculty abstractly. 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 class of languages definable within the grammar 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 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.

Grammatical theories, within these formalisms, are expressed as properties of a class of structures defined within a specific formal framework. Now, 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, RVSar]. 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] Such an approach raises a variety of questions about the nature of linguistic theories and the role of grammar formalisms in expressing them, questions that have consequences for both model-theoretic and traditional grammar-based approaches to theories of syntax.

In this paper, we explore some of these questions in the context of a model-theoretic reformalization of Generalized Phrase Structure Grammar (GPSG) [GKPS85].[2] While there is little current activity in GPSG, it is, for several reasons, an ideal choice as a subject for illuminating these issues. To a great extent, GPSG epitomizes the tradition of expressing linguistic principles in 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 procedural interpretations of the formalism. Finally, GPSG, at least in outline, has been a topic of a number of studies within the model-theoretic approach [BGMV93, BMV94, Kel93, Kra95] and these provide a basis for comparing the model-theoretic approach, as it has been developed so far, to the grammar-based approaches that preceded it.

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

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

[2]We will refer to [GKPS85] as GKP&S.

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[3]. We will refer to such aspects of linguistic theories as forming the Theoretic Level.[4] Note that equivalence at this level is stronger than the notions of either weak or strong generative capacity.

[3]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.

[4]We should note that the terms *descriptive* and /em 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.

There may be many ways of axiomatizing the same set of structures, each of which misses some of the generalizations captured by the others.[5]

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. As Blackburn, et al., note, the specific logical languages employed in [BGMV93] and [BMV94] 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

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

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 *inherited* and *instantiated* features, 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.[6] 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.

In this paper we focus on two aspects of a model-theoretic interpretation of GPSG that relate to these issues. On the descriptive level, we provide a treatment of Feature Specification Defaults that is considerably simplified, without the procedural character of the treatment of GKP&S. On the theoretic level, we capture the Exhaustive Constant Partial Ordering Property (ECPO), the principle underlying ID/LP format, explicitly as a property characterizing a class of sets of trees. Together with the other components of our re-interpretation, these definitions demonstrate that, at least in the case of GPSG, it is possible to give a model-theoretic account of linguistic theories at both the descriptive and theoretic level, and to do so in a reasonably uniform manner.

The fact that it can be difficult to capture the consequences of a grammar formalism is actually not so much an issue for the logical interpretation, as it is, or ought to be, an issue for the original theory. Such difficulties arise primarily because it is difficult to pin down, precisely, what these consequences are. Each of the treatments we discuss in this paper illuminates such an issue. In the case of FSDs, we argue that much of the apparent complexity of this notion in GKP&S is due to its expression in terms not far removed from generation. The model-theoretic definition we give is much simpler and, consequently, it is much easier to understand the effect of these defaults and the nature of the principles that they can express. In the case of ECPO we uncover an even stronger point. We will show that ECPO is only meaningful relative to a specification of the set of categories to which it applies, and that if this set of categories is taken

[6]We should note that 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.

to be the set of all categories in the sense of GKP&S, then ECPO will be violated in many of the sets in which it is presumed to hold. We suggest a reasonable alternative class of categories relative to which ECPO holds in the expected cases, but the point is not to patch the theory, but rather to point out that that any linguistic theory invoking ECPO is incomplete unless it specifies the set of categories to which it applies. One of the criticisms of model-theoretic treatments of grammar formalisms is that they tend to concentrate more on establishing that formalisms can be captured in this way than on establishing why one might want to do so. 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.[7]

Finally, the model-theoretic approach to syntax, along with all approaches that adopt relatively unrestricted formalisms, loses the clear language-theoretic complexity results that derive from formalisms with restricted generative capacity. One of the claims that is fundamental to the GPSG account of language is that human languages, or at least English, are context-free (in a mildly generalized sense). While the validity of this assertion, particularly in its universal form, is controversial, and even the linguistic relevance of such claims is open to debate, one can hardly claim to have captured GPSG without treating it. This provides a final rationale for our choice of GPSG as a case study. The formalism we employ, $L^2_{K,P}$, is a monadic second-order logic for reasoning about trees. A key virtue of this logic is that, while the signature is quite natural for expressing linguistically interesting constraints on the structure of trees, the class of sets of finite trees that are definable within it is nearly exactly the class of sets of trees definable in GPSG. 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.

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

[7]Again, that these issues are mooted by the fact that GPSG is no longer in active development is not lost on us. The point of raising them is not so much to advance the state of GPSG, as it is to demonstrate the way in which a linguistic theory can be informed by its model-theoretic interpretation.

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. 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.[8] While the underlying backbone of a GPSG grammar has the form of Context-Free Grammar, albeit one specified in ID/LP 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

[8]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].

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.[9] 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

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

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.

3. GPSG without Phrase-Structure Rules

In the remainder of this paper we sketch a few components of a model-theoretic reformalization of GPSG. We focus first on the descriptive level, sketching, in Section 4, our re-interpretation of FSDs. We then turn to the theoretic level in Section 5, and discuss our logical definition of the ECPO property. 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 ($\triangleleft^*$), immediate domination ($\triangleleft$), 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:

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

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.[10] 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.[11] We can think of a tree in this context as an assignment of labels

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

[11]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.

($\boldsymbol{P}$) to nodes in a subset of $\mathcal{N}_\omega$. Equivalently, we can interpret the labels as predicate variables, in which case the 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.[12] 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 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.

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

4. The Descriptive Level—Feature Specification Defaults

In this section we will focus on GPSG at the descriptive level. Our goal is a purely declarative formalization of the notion of Feature Specification Defaults. In order to achieve this, though, we will sketch most aspects of a complete formalization of GPSG at the descriptive level. 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 $\rho^0 : F \to \mathcal{P}(A)$ 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., stings 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 $\boldsymbol{C}$.[13]

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

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

For each $C \in \boldsymbol{C}$ we define a predicate picking out those nodes labeled with (an extension of) the corresponding category:[14]

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

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 occuring 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)

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

[13] 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.

[14] 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.

So each C picks out the set of nodes assigned to category C, each C^T 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

$$[\mathrm{ACC}](x) \text{ will be understood as } \langle \mathrm{CASE}, \mathrm{ACC} \rangle (x).$$

Similarly

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

and

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

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 $\vec{y}$.

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

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.

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

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

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

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.[15] 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*[16] of the grammar. Let $\mathbf{ID}_G$ be the set of all local contexts for the grammar G. Since the infinite 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 occuring 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 $\mathbf{ID}_G$:

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

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"

[15] See [Rog] for details.

[16] 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.

(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

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

where

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

Similarly,

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

translates as

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

4.4. **Universal Feature Instantiation 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 occuring 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:[17]

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

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.[18] 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.** FSDs provide default values for features which apply only if they are not overridden by the requirements of some other component of the grammar. The aspect of GKP&S's definition of GPSG that has the most dynamic flavor is their treatment of FSDs, so much so, in fact, that these are typically identified as an aspect of GPSG which is fundamentally incompatible with a declarative definition. For GKP&S, "some other component" includes just the ID rules (as expanded by the metarules), FCRs, and the FFP, CAP, and HFC. The ID rules and FCRs have only a local effect (i.e., the effect is limited to a local tree). The FFP, CAP, and HFC, on the other hand, can propagate those effects throughout the tree. The key point to note, though, is that FFP, CAP, and HFC cannot override FSDs by themselves. They simply require certain features on nodes that are related in certain ways to co-vary. While they can propagate values overriding FSDs from one local tree to another, every such value must be required somewhere in the tree by an ID rule or an FCR. In this way, all violations of FSDs must ultimately be justified by an ID rule or an FCR. The justifying configuration, however, may well not occur in the local tree.

[17] 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.

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

GKP&S defines a property PRIVILEGED which distinguishes the case when a node is prohibited from taking a default feature value by the requirements of the ID rules and FCRs as propagated by the FFP, CAP, and HFC. The definition is fairly intricate and is based on identifying features within a local tree that co-vary. Here we can exploit our ability, in $L^2_{K,P}$, to quantify over sets of nodes to provide a more direct definition of PRIVILEGED. The fundamental idea is that a node will be privileged to *not* take a feature f iff it is either required to take an incompatible feature by the ID rules or FCRs or it is included in a sequence of nodes that are pairwise required to co-vary on f by FFP, CAP, or HFC, that also includes a node that is required to carry an incompatible feature by the ID rules or FCRs. Thus f cannot appear on the privileged node without violating the requirements of the ID rules or FCRs either directly or by the propagation of that feature through the sequence of nodes.

Define $\text{Propagate}_f(x, y)$ to hold for every pair of nodes that are required to co-vary on the feature f by FFP, CAP, or HFC. Furthermore, restrict it to hold only if x either properly dominates or precedes y (thus making it an asymmetric relation). We can pick out nodes that are prohibited from taking the feature f with the negation of the predicate $\text{Free}_f(x)$ from the last section.[19] Let

$$
\begin{array}{l}
\text{Privileged}_f(x) \equiv \\
\quad (\exists y)\ [\neg\text{Free}_f(y)\wedge \\
\qquad (\exists S)\ [S(x) \wedge S(y)\wedge \\
\qquad\quad ((\forall z)[S(z) \rightarrow (z \approx x \vee (\exists z')[S(z') \wedge \text{Propagate}_f(z, z')])\wedge \\
\qquad\qquad\qquad (z \approx y \vee (\exists z')[S(z') \wedge \text{Propagate}_f(z', z)])] \vee \\
\qquad\quad \text{——}f \textbf{ propagates up to } x \textbf{ from } y \\
\qquad\quad (\forall z)[S(z) \rightarrow (z \approx x \vee (\exists z')[S(z') \wedge \text{Propagate}_f(z', z)])\wedge \\
\qquad\qquad\qquad (z \approx y \vee (\exists z')[S(z') \wedge \text{Propagate}_f(z, z')])] \\
\qquad\quad \text{——}f \textbf{ propagates down to } x \textbf{ from } y \\
\qquad\quad) \\
\qquad] \\
\quad].
\end{array}
$$

Since S may well be singleton (with $x \approx y$), $\text{Privileged}_f(x)$ will also hold in the case that x itself is not free to take the value f. One can define

[19]This will also identify nodes that are prohibited from taking f by the action of FFP or CAP in the local tree, but this is merely redundant. It is worth noting that the notion of being free to take a value for the purposes of HFC is defined, in GKP&S, only in terms of the local trees, ignoring long distance effects of FFP and CAP, whereas the notion of being free to take a value for the purposes of FSDs is specifically intended to account for those long distance effects.

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

These, then, let us capture the FSDs of GKP&S. For the default $[-\text{INV}]$, for instance, we get

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

For $[\text{BAR}\,0] \supset\!\sim [\text{PAS}]$, we get

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

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.

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 $\boldsymbol{P}$[20] 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 ECPO Property

The previous section explains how, given a GPSG grammar G, we can construct an $L^2_{K,P}$ formula Φ_G that captures the theory expressed by G at the descriptive level. Thus there is a class of $L^2_{K,P}$ formulae that correspond to GPSG grammars. In this section we explain how to capture a theoretic level aspect of GPSG 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

[20]Note that a more consistent notation would be $\Phi_G(\vec{\text{P}})$ 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.

properties of the sets of models defined at the descriptive level.[21] 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.[22] Effectively we translate GKP&S's restricted class of grammars to a restricted class of definitions of sets of trees. The particular component of the theoretical level of GPSG that we explore is the Exhaustive Constant Partial Ordering Property.

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.

We begin by defining an auxiliary predicate. Let

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

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.

[21]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).

[22]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.

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

ECPO:

$$\bigwedge_{C_2^T, C_3^T \in \boldsymbol{C}^T} \Bigg[\bigvee_{C_1^T \in \boldsymbol{C}^T} \left[\begin{array}{c} (\exists \boldsymbol{P})\big[\Phi_G(\boldsymbol{P}) \wedge (\exists x,y,z)[C_1^T(x) \wedge \tau_{2,3}(x,y,z)]\big] \wedge \\ (\forall \boldsymbol{P})\big[\Phi_G(\boldsymbol{P}) \rightarrow (\forall x,y,z)[C_1^T(x) \wedge \tau_{2,3}(x,y,z) \rightarrow z \not\prec y]\big] \end{array} \right] \longrightarrow \quad (\forall \boldsymbol{P})\big[\Phi_G(\boldsymbol{P}) \rightarrow (\forall x,y,z)[\tau_{2,3}(x,y,z) \rightarrow z \not\prec y]\big] \Bigg]$$

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:

$$\begin{array}{ll} [-\mathrm{N}, +\mathrm{V}, +\mathrm{SUBJ}, \mathrm{BAR2}] & \longrightarrow \\ & \qquad [+\mathrm{N}, -\mathrm{V}, \mathrm{BAR2}]\ [-\mathrm{N}, +\mathrm{V}, -\mathrm{SUBJ}, \mathrm{BAR2}] \\ [-\mathrm{N}, +\mathrm{V}, -\mathrm{SUBJ}, \mathrm{BAR2}] & \longrightarrow \quad [-\mathrm{N}, +\mathrm{V}, \mathrm{BAR0}]\ [+\mathrm{N}, -\mathrm{V}, \mathrm{BAR2}] \\ [-\mathrm{N}, +\mathrm{V}, -\mathrm{SUBJ}, \mathrm{BAR2}] & \longrightarrow \\ \multicolumn{2}{r}{[-\mathrm{N}, +\mathrm{V}, \mathrm{BAR0}]\ [+\mathrm{N}, -\mathrm{V}, \mathrm{BAR2}]\ [-\mathrm{N}, +\mathrm{V}, -\mathrm{SUBJ}, \mathrm{BAR2}]} \end{array}$$

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], $[+\mathrm{N}, -V]$, and $[-\mathrm{N}, +V]$ as well, and here we find that $[+\mathrm{N}, -V]$ always precedes $[-\mathrm{N}, +V]$ 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 occuring 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

We have demonstrated how one can capture components of the theory of syntax expressed in GPSG by defining properties of sets of trees licensed by GPSG grammars within $L^2_{K,P}$, and how, in doing so, one gains considerable clarity about the nature and consequences of these components. In the case of ECPO, in fact, this approach has uncovered a gap in the GKP&S definition of GPSG—one that has apparently gone unnoticed despite the intense scrutiny to which the definition, and in particular the claim of ECPO as a principle of universal grammar, has been subjected. We feel that the possibility of such results is a powerful justification for model-theoretic explorations of grammar-based syntactic theories. On the other hand, such model-theoretic explorations have heretofore focused simply on the set of models licensed by a given grammar; they have been limited to the descriptive level. While such studies still have much to offer the original theory—as witnessed by our re-interpretation of FSDs—much of the potential value of the approach will be missed unless this focus is broadened beyond just the theories of individual languages to encompass theories of natural language 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.

[Rog] James Rogers. What does a grammar formalism say about a language? In Preparation.

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

[RVSar] James Rogers and K. Vijay-Shanker. Obtaining trees from their descriptions: An application to tree-adjoining grammars. *Computational Intelligence*, To appear.

[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.

Institute for Research in Cognitive Science, University of Pennsylvania, Suite 400C, 3401 Walnut Street, Philadelphia, PA 19104, USA

E-mail address: jrogers@linc.cis.upenn.edu