May 1996

What Does a Grammar Formalism Say About a Language?

James Rogers
University of Pennsylvania, jrogers@linc.cis.upenn.edu

University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS 96-10.

What Does a Grammar Formalism Say About a Language?

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. This approach raises a number of questions about the nature of linguistic theories and the role of grammar formalisms in expressing them.

We argue here that the crux of what theories of syntax have to say about language lies in the abstract properties of the sets of structures they license. This is the level that is most directly connected to the empirical basis of these theories and it is the level at which it is possible to make meaningful comparisons between the approaches. From this point of view, grammar formalisms, or (formal frameworks) are primarily means of presenting these properties. Many of the apparent distinctions between formalisms, then, may well be artifacts of their presentation rather than substantive distinctions between the properties of the structures they license. The model-theoretic approach offers a way in which to abstract away from the idiosyncrasies of these presentations.

Having said that, we must distinguish between the class of sets of structures licensed by a linguistic theory and the set of structures licensed by a specific instance of the theory—by a grammar expressing that theory. Theories of syntax are not simply accounts of the structure of individual languages in isolation, but rather include assertions about the organization of the structure of human languages in general. These universal aspects of the theories present two challenges for the model-theoretic approach. First, they frequently are not properties of individual structures, but are rather properties of sets of structures. Thus, in capturing these model-theoretically one is not defining sets of structures but is rather defining classes of sets of structures; these are not first order properties. Secondly, the universal aspects of linguistic theories are frequently not explicit, but are consequences of the nature of the formalism that embodies the theory. In capturing these one must develop an explicit axiomatic treatment of the formalism. This is both a challenge and a powerful beneft of the approach. Such re-interpretations tend to raise a variety of issues that are often overlooked in the original formalization.

In this report we examine these issues within the context of a model-theoretic reinterpretation of Generalized Phrase-Structure Grammar. While there is little current active research on GPSG, it provides an ideal laboratory for exploring these issues. First, the formalism of GPSG is expressly intended to embody a great deal of the accompanying linguistic theory. Thus it provides a variety of opportunities for examining principles expressed as restrictions on the formalism from a model-theoretic point of view. At the same time, the fact that these restrictions embody universal grammar principles provides us with a variety of opportunities to explore the way in which the linguistic theory expressed by a grammar can transcend the mathematical theory of the structures it licenses. Finally, GPSG, although defined declaratively, is a formalism with restricted generative capacity, a characteristic more typical of the earlier procedural formalisms. As such, one component of the theory it embodies is a claim about the language-theoretic complexity of natural languages. Such claims are difficult to establish for any of the constraint-based approaches to grammar. We can show, however, that the class of sets of trees that are definable within the logical language we employ in reformalizing GPSG is nearly exactly the class of sets of trees definable within the basic GPSG formalism.

Thus we are able to capture the language-theoretic consequences of GPSGs restricted formalism by employing a restricted logical language.

Comments

University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS 96-10.

Institute for Research in Cognitive Science

What Does a Grammar Formalism Say About a Language?

James Rogers

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

May 1996

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

IRCS Report 96--10

What Does a Grammar Formalism Say About a Language?

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

May 14, 1996

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. This approach raises a number of questions about the nature of linguistic theories and the role of grammar formalisms in expressing them.

We argue here that the crux of what theories of syntax have to say about language lies in the abstract properties of the sets of structures they license. This is the level that is most directly connected to the empirical basis of these theories and it is the level at which it is possible to make meaningful comparisons between the approaches. From this point of view, grammar formalisms (or formal frameworks) are primarily means of presenting these properties. Many of the apparent distinctions between formalisms, then, may well be artifacts of their presentation rather than substantive distinctions between the properties of the structures they license. The model-theoretic approach offers a way in which to abstract away from the idiosyncrasies of these presentations.

Having said that, we must distinguish between the class of sets of structures licensed by a linguistic theory and the set of structures licensed by a specific instance of the theory—by a grammar expressing that theory. Theories of syntax are not simply accounts of the structure of individual languages in isolation, but rather include assertions about the organization of the structure of human languages in general. These universal aspects of the theories present two challenges for the model-theoretic approach. First, they frequently are not properties of individual structures, but are rather properties of sets of structures. Thus, in capturing these model-theoretically one is not defining sets of structures but is rather defining classes of sets of structures; these are not first-order properties. Secondly, the universal aspects of linguistic theories are frequently not explicit, but are consequences of the nature of the formalism that embodies the theory.

In capturing these one must develop an explicit axiomatic treatment of the formalism. This is both a challenge and a powerful benefit of the approach. Such re-interpretations tend to raise a variety of issues that are often overlooked in the original formalization.

In this report we examine these issues within the context of a model-theoretic reinterpretation of Generalized Phrase-Structure Grammar. While there is little current active research on GPSG, it provides an ideal laboratory for exploring these issues. First, the formalism of GPSG is expressly intended to embody a great deal of the accompanying linguistic theory. Thus it provides a variety of opportunities for examining principles expressed as restrictions on the formalism from a model-theoretic point of view. At the same time, the fact that these restrictions embody universal grammar principles provides us with a variety of opportunities to explore the way in which the linguistic theory expressed by a grammar can transcend the mathematical theory of the structures it licenses. Finally, GPSG, although defined declaratively, is a formalism with restricted generative capacity—a characteristic more typical of the earlier procedural formalisms. As such, one component of the theory it embodies is a claim about the language-theoretic complexity of natural languages. Such claims are difficult to establish for any of the constraint-based approaches to grammar. We can show, however, that the class of sets of trees that are definable within the logical language we employ in reformalizing GPSG is nearly exactly the class of sets of trees definable within the basic GPSG formalism. Thus we are able to capture the language-theoretic consequences of GPSG's restricted formalism by employing a restricted logical language.

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.

The paper is structured as a case study in application of a model-theoretic approach to 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 constructing them from the ground up. 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 within 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 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

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

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

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. 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.

[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 *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.

[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.

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 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.

Our model-theoretic reinterpretation explicitly incorporates those components of GPSG that, like metarules and ID/LP format, operate at the theoretic level. Further, we extricate the notions of inherited, instantiated and free features from their apparent dependence on a procedural interpretation of the grammar. We demonstrate, then, that, at least for 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. 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 at all apparent in their original form. Our reinterpretation of GPSG uncovers a number of such issues, issues which we believe to be significant for GPSG itself, no matter how formalized. 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. To this end, we have tried to emphasize these issues as they arise in our formalization.[7]

Finally, the model-theoretic approach to syntax, along with all approaches that adopt

[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.

[7] Again, that the significance of these issues is limited by 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.

relatively unrestricted formalisms, loses the clear language-theoretic complexity results obtainable 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 hypotheses of this sort 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 with the key virtue 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 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 logical 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 pu-

[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].

> tative 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 subcategories 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 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 model-theoretic reformalization of GPSG, first, in Section 4, at the descriptive level, and then, in Section 5, at the theoretic level. 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.

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

As a simple example of the kinds of assertions this allows one to make consider the formula:

$$\text{Stratification}(X) \equiv (\forall x, y)[x \triangleleft y \rightarrow (X(x) \leftrightarrow \neg X(y))].$$

This defines a property of sets: a set X is a stratification iff whenever x and y are individuals related by parent one is in X iff the other is not—in other words, X is a stratification iff it consists of every other generation of nodes in a tree. We can use this to define a relation between nodes:

$$\text{Even}(x, y) \equiv (\exists X)[\text{Stratification}(X) \wedge X(x) \wedge X(y)].$$

This holds for a pair of nodes iff the difference between their depths in the tree is even.[10] We will continue to follow the conventions employed in these 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 labeled tree in this context as an assignment of labels ($\boldsymbol{P}$) to the 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.[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

[10] There is no obvious linguistic significance to this relation. It is, however, easy to show that Even(x, y) is not first-order definable.

[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.

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

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.

This reinterpretation of GPSG raises a variety of issues which have not, to our knowledge, previously been delineated. While we do not presume to resolve these, we do offer some potential modifications to theory which account for them. For instance:

- We offer a simplified approach to the definition of the notion of a privileged feature with respect to FSDs and a more uniform treatment of inherited/free/privileged features.
- We explore the relationship between the ID/LP format of the grammar and the ECPO property and note the significance of the question of which categories ECPO applies to and of whether it applies to the sets of trees licensed by the grammar as a whole or only to those licensed by the ID/LP component in isolation.
- We revisit the question of restricting the application of metarules to prevent them from generating infinite grammars. While we don't require any particular approach, the existence of such restrictions fall out as a consequence of our formalization in the sense that the formulae with which we encode metarules will be unsatisfiable if those metarules generate infinite grammars.
- We raise the question of why FCRs—which state generalizations about the co-occurrence of features on nodes—should have consequences for FFP and CAP. That is, why should the issue of whether a feature is propagated by these principles depend on whether the grammar explicitly assigns the feature with an ID rule or assigns it as a consequence of such a generalization.
- In considering this, we suggest that perhaps features specified by FCRs should not be distinguished from those specified by ID rules. To accommodate the issues leading to the distinction in GKP&S we suggest a modified form for STM1 along with an account of the propagation of SLASH features in the local trees it licenses that is based in the CAP.
- Finally, we consider the restriction of the application of Kleene closure to single categories, and (informally) suggest a strengthening of ECPO which would imply such a restriction as a consequence.

While we would like to think that these points help to illuminate the theory of GPSG, we should emphasize again that we believe their significance is not so much what they actually say about that theory as it is the fact that they only become apparent in the light of its model-theoretic interpretation. The reconstruction of the theory within this framework does more than simply provide an alternative formalization, it offers considerable clarification of the details of the theory and their consequences.

4 The Descriptive Level—The Set of Trees Licensed by GKP&S

In this section we will focus on GPSG at the descriptive level. Our goal is a purely (and clearly) 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 a 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 non-empty 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

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

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

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)

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

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.

Extensions and unification

For any two sets $C_1, C_2 \in \boldsymbol{C}$

$$C_1 \sqsubseteq C_2 \Leftrightarrow C_1 \subseteq C_2.$$

That is, C_2 extends C_1 iff the set of paths for which it is defined is an extension of the set of paths for which C_1 is defined. With our definition of the formula $C(x)$, this becomes

$$C_1 \sqsubseteq C_2 \Leftrightarrow (\forall x)[C_2(x) \rightarrow C_1(x)].$$

In other words, C_2 extends C_1 iff the set of nodes assigned to category C_2 is a subset of those assigned to category C_1.[15]

[14] There are several obvious ways to extend this. First, we might take categories to be sets of atomic formulae over 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.

[15] Note, in particular, that the subset relation reverses in going from the view of categories as a set of formulae (equivalently paths) to the view as a set of nodes.

A category C_0 is a unifier of categories C_1 and C_2 iff it extends them both. In terms of sets of literals, it is a unifier iff it is a superset of the union of C_1 and C_2. It is the *unification* (most general unifier) of C_1 and C_2 if it is exactly their union.[16]

$$C_1 \sqcup C_2 = C_1 \cup C_2.$$

Equivalently, C_0 is a unifier of C_1 and C_2 iff the set of nodes assigned to C_0 is a subset of the intersection of those assigned to C_1 and those assigned to C_2.[17]

$$C_0 = C_1 \sqcup C_2 \Leftrightarrow (\forall x)[C_0(x) \leftrightarrow (C_1(x) \land C_2(x))].$$

Abbreviations

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

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

and

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

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

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

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 and Metarules

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 $\vec{y}$.

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

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.

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

[16] In the presence of FCRs, at least, the union may not occur in C even when extensions of the union do. In such a case the minimal extensions with respect to subset are the most general unifiers. There is a unique mgu only if there is a unique minimal extension. We ignore this issue.

[17] Here the $\rightarrow$ direction does not hold if $C_1 \cup C_2 \notin C$. Note again that union of the sets of nodes in the categories corresponds to the intersection of features that define those categories.

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 noted above, these are principles of GPSG at the theoretic level. Their treatment, consequently, will be deferred until the next section. Here we understand them as simply specifying finite sets of additional ID rules, formed from the basic rules explicitly presented in the grammar. We will assume then, 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*[18] 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 $\mathbf{ID}_G$:

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

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" (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

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

[18] 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.

Similarly,

$$[\text{SUBCAT}] \prec\sim [\text{SUBCAT}]$$

translates as

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

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. These principles operate on both the descriptive and theoretic levels. First, they control the co-occurrence of certain sets of features on certain pairs of nodes in local trees. Beyond this, though, they make a claim that there are particular linguistically relevant sets of features that are governed in these ways. While GKP&S does not argue the FFP at this level to any great extent, the justification of CAP at this level occupies the bulk of its exposition in GKP&S, and the idea that some such set of Head features exists and is governed by something like HFC is central to the very notion of *head*. We are concerned, in this section, with the definition of these principles at the descriptive level. We will take up the theoretic level in Section 5.3.

The Foot Feature Principle

The Foot Feature Principle governs the propagation of a specific class of category-valued features through the tree. Fundamentally, it states that the value of these features for the parent of a local tree must be the unification of the values of these features on the children. Values that are stipulated by the licensing context of the local tree (*inherited features*), however, are exceptions to the principle (that is, features that are explicit in the ID rule that licenses the local tree need not propagate). The foot features occuring in GKP&S are just SLASH, WH, and RE. We can define the set of paths through foot features, then, as the set of all paths in $\mathbf{P}^T$ starting with a foot feature:

$$\mathbf{FOOT} = (\text{SLASH} + \text{WH} + \text{RE})F^*A \cap \mathbf{P}^T$$

(in which '+' denotes alternation). The basic agreement component of FFP simply requires that the a label in **FOOT** is assigned to a node iff it is assigned to one of its children:

$$(\forall x, \vec{y})[\text{Children}(x, \vec{y}) \rightarrow \bigwedge_{f \in \mathbf{FOOT}} [f(x) \leftrightarrow \bigvee_{y \in \vec{y}} [f(y)]]].$$

We still need to account for the exceptions to the FFP. These are the inherited features—configurations of features that are explicitly stipulated by the ID rules. We capture these as explicit exceptions for each of the features. The SLASH feature can be introduced by, among others,[19]

$$\begin{array}{rcl} \mathrm{A}^1 & \longrightarrow & \mathrm{H}[42],\ \mathrm{V}^2[\mathrm{INF}]/\mathrm{NP}[-\mathrm{NOM}] \\ \mathrm{VP}[+it] & \longrightarrow & \mathrm{H}[44],\ \mathrm{X}^2,\ \mathrm{S}[\mathrm{FIN}]/\mathrm{X}^2 \\ & \vdots & \end{array}$$

We can gather these and the other local contexts licensing SLASH into a predicate that is true of a node in the case that it has an inherited SLASH feature:

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

where the ID_n formulae are just the local contexts of the ID rules:

$$\begin{array}{rcl} \mathrm{ID}_{42}(y_0, y_1, y_2) & \equiv & \mathrm{Children}(y_0, y_1, y_2) \wedge \mathrm{A}^1(y_0) \wedge \mathrm{H}[42](y_1) \wedge \\ & & \mathrm{V}^2[\mathrm{INF}]/\mathrm{NP}[-\mathrm{NOM}](y_2) \\ \mathrm{ID}_{44}(y_0, y_1, y_2, y_3) & \equiv & \mathrm{Children}(y_0, y_1, y_2, y_3) \wedge \mathrm{VP}[+it](y_0) \wedge \mathrm{H}[44](y_1) \wedge \\ & & [\mathrm{BAR}\ 2](y_1) \wedge \mathrm{S}[\mathrm{FIN}]/[\mathrm{BAR}\ 2](y_3). \end{array}$$

It should be noted that GKP&S takes inherited features to be those occurring in the set of ID rules as expanded by the metarules but "prior to" application of FCRs.[20]

We can then formulate the FFP:

FFP :

$$(\forall x)[\ (\exists y)[x \triangleleft y] \rightarrow \bigwedge_{f \in \mathbf{FOOT}} [(f(x) \wedge \neg \mathrm{Inh}_f(x)) \leftrightarrow (\exists y)[f(y) \wedge \neg \mathrm{Inh}_f(y)]]].$$

—**The set of instantiated FOOT features of x is the union of the sets of instantiated foot features of its children.**

The Control Agreement Principle

The Control Agreement Principle governs the co-occurrence of features assigned to nodes in more or less typical agreement relationships. In GKP&S the control features (those to which the principle applies) are the AGR and SLASH features. There is a well developed theory, based in the semantic interpretation component of the grammar, of which nodes in a given local tree are required to agree and which of the control features they are required to

[19] Note that the agreement between the two occurrences of X^2 in the second of these rules is enforced not by the coincidence of the X (which is not a variable but rather a null category specification), but rather by the Control Agreement Principle.

[20] If features required by FCRs were taken to be instantiated then no slashes could be terminated by rules generated by STM1.

agree on. The reader is referred to GKP&S for details and justification of the theory. Here we will assume that agreeing nodes are picked out by the auxiliary predicates Controller(x) and Target(x). The relevant control feature is SLASH if it is an inherited feature of the controller, otherwise it is AGR. We will say

$$\begin{aligned}\mathrm{SCONTROL}(x) &\equiv \mathrm{Inh}_{[\mathrm{SLASH}]}(x) \wedge \neg[\mathrm{BAR}\ 0](x)\\ \mathrm{ACONTROL}(x) &\equiv [\mathrm{AGR}](x) \wedge \neg[\mathrm{SLASH}](x) \wedge \neg[\mathrm{BAR}\ 0](x).\end{aligned}$$

The restriction to non-Bar 0 categories occurs here in parallel to the treatment in GKP&S.

The particular features on which a controller must agree with its target are either the control features themselves or are the values of the control feature that are **HEAD** features but not **FOOT** features or are inherited **FOOT** features of the target. CAP itself breaks down into two clauses: one for the configuration in which the target has a controller as a sibling, in which case the relevant control feature is that of the target; and one for the configuration in which the target has no controller,[21] in which case agreement is required between the value of the control feature of the parent and the value of the control feature of the target (the relevant control feature may differ between these). We can construct a definition of CAP, then, as follows:

$$\begin{aligned}\mathrm{SlAgr}(x,y) \equiv\ & \bigwedge_{\substack{f\in\\ \mathrm{HEAD}\setminus\mathrm{FOOT}}} [[\mathrm{SLASH}\, f](x) \leftrightarrow (f(y) \vee [\mathrm{SLASH}\, f](y))] \ \wedge\\ & \text{—}\textbf{Value of SLASH on } x \textbf{ for HEAD-non-FOOT features}\\ & \quad\textbf{is unification of the value of those features on } y \textbf{ and}\\ & \quad\textbf{the value of SLASH on } y \textbf{ for those features.}\\ & \bigwedge_{f\in\mathrm{FOOT}} [[\mathrm{SLASH}\, f](x) \leftrightarrow (f(y) \wedge \mathrm{Inh}_f(y) \vee [\mathrm{SLASH}\, f](y))]\\ & \text{—}\textbf{Value of SLASH on } x \textbf{ for FOOT features is unification}\\ & \quad\textbf{of the value of the inherited FOOT features on } y \textbf{ and}\\ & \quad\textbf{the value of SLASH on } y \textbf{ for those features.}\end{aligned}$$

$$\begin{aligned}\mathrm{AgrAgr}(x,y) \equiv\ & \bigwedge_{\substack{f\in\\ \mathrm{HEAD}\setminus\mathrm{FOOT}}} [[\mathrm{AGR}\, f](x) \leftrightarrow (f(y) \vee [\mathrm{AGR}\, f](y))] \ \wedge\\ & \text{—}\textbf{Value of AGR on } x \textbf{ for HEAD-non-FOOT features}\\ & \quad\textbf{is unification of the value of those features on } y \textbf{ and}\\ & \quad\textbf{the value of AGR on } y \textbf{ for those features.}\\ & \bigwedge_{f\in\mathrm{FOOT}} [[\mathrm{AGR}\, f](x) \leftrightarrow (f(y) \wedge \mathrm{Inh}_f(y) \vee [\mathrm{AGR}\, f](y))]\\ & \text{—}\textbf{Value of AGR on } x \textbf{ for FOOT features is unification}\\ & \quad\textbf{of the value of the inherited FOOT features on } y \textbf{ and}\\ & \quad\textbf{the value of AGR on } y \textbf{ for those features.}\end{aligned}$$

$$\begin{aligned}\mathrm{SibAgr}(x,y) \equiv\ & (\mathrm{SCONTROL}(x) \rightarrow \mathrm{SlAgr}(x,y)) \wedge (\mathrm{ACONTROL}(x) \rightarrow \mathrm{AgrAgr}(x,y))\\ & \text{—}y \textbf{ controls a sibling } x.\end{aligned}$$

[21] More correctly, in which the target is a predicative category with no controller, but we will assume this is a component of the distribution of Target(x).

$$
\begin{aligned}
&\mathrm{ParAgr}(x,y) \equiv \\
&\quad \mathrm{SCONTROL}(x) \wedge \mathrm{SCONTROL}(y) \wedge (\bigwedge_{f \in \mathbf{P}^T} [[\mathrm{SLASH}\, f](x) \leftrightarrow [\mathrm{SLASH}\, f](y)]) \vee \\
&\quad \mathrm{SCONTROL}(x) \wedge \mathrm{ACONTROL}(y) \wedge (\bigwedge_{f \in \mathbf{P}^T} [[\mathrm{SLASH}\, f](x) \leftrightarrow [\mathrm{AGR}\, f](y)]) \vee \\
&\quad \mathrm{ACONTROL}(x) \wedge \mathrm{SCONTROL}(y) \wedge (\bigwedge_{f \in \mathbf{P}^T} [[\mathrm{AGR}\, f](x) \leftrightarrow [\mathrm{SLASH}\, f](y)]) \vee \\
&\quad \mathrm{ACONTROL}(x) \wedge \mathrm{ACONTROL}(y) \wedge (\bigwedge_{f \in \mathbf{P}^T} [[\mathrm{AGR}\, f](x) \leftrightarrow [\mathrm{AGR}\, f](y)])
\end{aligned}
$$

—y **controls a child** x**.**

CAP :

$$
\begin{aligned}
(\forall x)[\ & \mathrm{Target}(x) \rightarrow \\
&\quad (\forall y)[\mathrm{Sibling}(x,y) \wedge \mathrm{Controller}(y) \rightarrow \mathrm{SibAgr}(x,y)] \wedge \\
&\quad (\neg(\exists y)[\mathrm{Sibling}(x,y) \wedge \mathrm{Controller}(y)] \rightarrow \\
&\qquad (\exists y)[y \triangleleft x \wedge \mathrm{ParAgr}(x,y)]) \\
&].
\end{aligned}
$$

Note that this is logically equivalent to the conjunction of a (finite) set of biconditionals that identify each of the local contexts in which CAP holds and require the relevant features to co-occur on the relevant nodes in that context.

The Head Feature Convention

The fundamental distinctions between the HFC and the FFP (other than the set of features it governs) are that a node is required to be labeled with the generalization (intersection) of the head features of the heads expanding it (rather than the unification required by the FFP) and that the HFC is overridden by not only by the ID rules (as expanded by the metarules) but also by the FCRs, FFP and CAP. In GKP&S, HFC is restricted to apply only to *free* feature specifications—those that may label a node consistent with these other components of the grammar. Here we will assume, for each label f, a predicate $\mathrm{Free}_f(x)$ which is true at x iff x can be labeled f consistent with the ID rules, the FCRs, FFP, and CAP. The definition of Free_f can be carried out in a manner similar to that of Inh_f (or rather something like its negation).[22]

[22] In GKP&S, *free* is defined in terms of the possible expansions of the licensing context. We will approach HFC in this way when we treat the Universal Instantiation Principles on the theoretical level. Here we are concerned only with its consequences for the set of trees licensed by the grammar.

With this we can capture the HFC in direct parallel with the definition in GKP&S:

HFC :

$(\forall x)[\ (\exists x)[x \triangleleft y \wedge \mathrm{H}(y)] \rightarrow$

—**Whenever** x **has a head child**

$(\ [BAR](x) \wedge (\forall y)[(x \triangleleft y \wedge \mathrm{H}(y)) \rightarrow [BAR](y)] \wedge$

—x **and each of its head children has a bar level**

$(\forall y)[(x \triangleleft y \wedge \mathrm{H}(y)) \rightarrow \bigwedge_{f \in \mathbf{HEAD}}[(f(x) \wedge \mathrm{Free}_f(y)) \rightarrow f(y)]] \wedge$

—**Every HEAD feature of** x **that is free for a head child appears on that head child**

$\bigwedge_{f \in \mathbf{HEAD}}[(\ (\exists y)[x \triangleleft y \wedge \mathrm{H}(y) \wedge f(y)] \wedge$
$(\forall y)[(x \triangleleft y \wedge \mathrm{H}(y)) \rightarrow f(y)] \wedge \mathrm{Free}_f(x)) \rightarrow f(x)]$

—**Every HEAD feature appearing on all head children that is free for** x **appears on** x

)

].

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. 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.

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 $\mathrm{Propagate}_f(x, y)$ to hold for every pair of nodes that are required to co-vary on the feature f by FFP, CAP, or HFC. Note that Propagate is symmetric. We can pick out nodes that are prohibited from taking the feature f with the negation of the predicate

$\mathrm{Free}_f(x)$ from the last section.[23] The set of nodes that are privileged wrt f includes all nodes that are not Free for f as well as any node connected to such a node by a sequence of $\mathrm{Propagate}_f$ links. We, in essence, define this inductively. $\mathrm{P}'_f(X)$ is true of a set iff it includes all nodes not Free for f and is closed wrt $\mathrm{Propagate}_f$. $\mathrm{PrivSet}_f(X)$ is true of the smallest such set.

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

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

There are two things to note about this definition. First, in any tree there is a unique set satisfying $\mathrm{PrivSet}_f(X)$ and this contains exactly those nodes not Free for f or connected to such a node by $\mathrm{Propagate}_f$. 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 $\mathrm{PrivSet}_f$.[24]

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

One can define $\mathrm{Privileged}_{\neg f}(x)$ 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:

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

For [BAR 0] $\supset\!\sim$ [PAS], we get:

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

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.

[23] 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.

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

4.5 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 $\boldsymbol{P}$[25] 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 Linguistic Theory of GKP&S

The previous section shows 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.[26] 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.[27] 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.

[25] Note that a more consistent notation would be $\Phi_G(\vec{\mathrm{P}})$ explicitly treating 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.

[26] 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).

[27] 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.

In stating the ECPO property independently of the underlying grammar, we do not want to refer to the actual set of categories occuring in the local contexts of the ID component of Φ_G, since we do not want to assume Φ_G has such a structure. For every set of trees, however, there will be a set of local trees occuring in those trees that are minimal in the sense that there are no other local trees occurring in the set that are labeled with less specific categories but are otherwise identical. Every local tree occurring in a tree in the set, then, will be an instance of one of these minimal local trees. These minimal local trees correspond roughly to the ID rules of a GPSG grammar and we will generally refer to them where GKP&S refers to the rules. The categories labeling their nodes are the categories to which ECPO must apply in any case. We adopt a stronger interpretation of ECPO, however, and require it to apply to all extensions of those categories that actually occur in the trees. This is accomplished simply by stating ECPO in terms of $\boldsymbol{C}^T$ since it is vacuously true for all categories in $\boldsymbol{C}^T$ that do not occur in the set of trees.

We begin by defining an auxiliary predicate. Let

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

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:

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

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:

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

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], $[+\mathrm{N}, -V]$, 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.

5.2 Metarules

Metarules in GPSG are a set of schemata which map lexical ID rules (those in which a Bar 0 head occurs on the right hand side) to lexical ID rules. An example is the *Passive Metarule*:

$$\begin{array}{c} \textit{Passive Metarule} \\ \mathrm{VP} \longrightarrow W,\ \mathrm{NP} \\ \Downarrow \\ \mathrm{VP[PAS]} \longrightarrow W,\ (\mathrm{PP}[by]) \end{array}$$

This says that for every ID rule licensing the expansion of VP to an NP along with other material, there is an ID rule licensing the expansion of a passive-form sentence (VP[PAS]) in the same way with the omission of the NP (and optionally including a *by* PP). Crucially, these are not transformations. They do not apply to the structures licensed by the grammar, but rather to the grammar itself. In addition, they are restricted to apply in a non-iterative fashion—no metarule may apply to an ID rule that was generated by that same rule. These restrictions on the applicability of metarules have the effect of limiting them to "rule-collapsing conventions" [GKPS85, pg. 66]. Their formal effect is only as a mechanism for abbreviating the presentation of the grammar; they do not extend its generative power. Linguistically, on the other hand, they serve to express closure conditions on the grammar, and thus on the set of trees it defines. They state that, if the grammar licenses certain local trees then it must necessarily license certain other local trees as well. Such closure properties restrict the class of languages definable by GPSG grammars that employ the metarule. As such, the metarules are statements of linguistic principles that are shared, at least, by certain classes of languages. Some presumably have the status of principles of Universal Grammar on a par with the ECPO principle.

As with our treatment of the ECPO property, we have already captured the effect of the metarules at the descriptive level simply by basing Φ_G on the expanded set of ID rules of G. This guarantees that Φ_G defines the correct set of trees, but it loses the generalization expressed by the metarule. We reintroduce that generalization by expressing it as a property of the definition. In doing so, we will need to reference explicitly the set of categories instantiated in a given local tree. In the absence of iteration on the right hand side of rules this is straightforward. Since the size of the local trees will be bounded and there are boundedly many features the categories can realize, we can identify each local tree with a finite conjunction. For simplicity of presentation, we will treat this finite case here, and return to the issue of generalizing for unbounded rule schema in Section 5.4. Under this assumption, we can state the passive metarule as follows:

Passive metarule:

$$\begin{array}{l}(\exists \boldsymbol{P})[\Phi_G(\boldsymbol{P}) \wedge (\exists x, y, \vec{z})[\mathrm{Children}(x, y, \vec{z}) \wedge \mathrm{VP}(x) \wedge \mathrm{NP}(y) \wedge \bigwedge_{z \in \vec{z}}[\phi_z(z)]]] \rightarrow \\ \qquad (\exists \boldsymbol{P})[\Phi_G(\boldsymbol{P}) \wedge (\exists x, \vec{z})[\mathrm{Children}(x, \vec{z}) \wedge \mathrm{VP[PAS]}(x) \wedge \bigwedge_{z \in \vec{z}}[\phi_z(z)]]]\end{array}$$

This is a finite schema in which $\vec{z}$ takes on sequences of variables of length bounded by the maximum branching of the rules in G. The ϕ_z range through all categories in $\mathbf{C}^T$. In words, this states that whenever Φ_G is satisfied by some tree including a local tree in which VP expands to an NP and some specific sequence of other nodes, then there is also a tree satisfying Φ_G in which there is a local tree in which VP[PAS] expands into just that sequence of other nodes (we ignore the optional *by*-phrase). As with ECPO this property restricts us to a decidable class of definitions which capture the class of languages in which the generalization expressed by the metarule is valid. Also as with ECPO, in this formulation we have a closure property of the sets of trees capturing those languages that corresponds directly to the linguistic content of the metarule.

Note that, as given, the encodings of the metarules apply to the the full set of structures licensed by the definition—directly contradicting the restrictions on applicability of metarules in GPSG. Specifically, there is no restriction corresponding to those limiting application of metarules non-iteratively and only to lexical ID rules. Both of these are linguistically important restrictions. We can capture the first of these, which has specific empirical consequences in the analysis of unbounded dependencies, simply by extending the translation of the metarule to require a Bar zero head child in the antecedent of the implication. The second restriction—that no metarule may be applied to its own output—is required to assure that the expanded grammar is finite. Note that if any metarule in a given grammar *can* apply productively to its own output (or to the result of applying other metarules to that output) then the grammar will license local trees that match the pattern of the metarule without licensing the local trees the metarule derives from them. In particular, it will include trees derived by a single application of the metarule but none derived by subsequent applications. Thus the statement of the metarule as a closure property on the sets of models may be too strong. At the descriptive level we get that Φ_G licenses a set of trees corresponding to a grammar expanded under the restriction to non-iterative application simply by observing the restriction in generating Φ_G. But Φ_G will not satisfy the closure property. In fact, no finite Φ_G could. Thus the encoding of that metarule will be unsatisfiable as a property of definitions.

In examining the metarules proposed in GKP&S, one finds that only the passive metarule and the Slash Termination Metarule 1 (STM1) can apply to their own output. The linguistic principle that is expressed by the passive metarule, arguably, is that every VP that is not in passive form has a corresponding S that is in passive form. Thus we might re-encode the passive metarule as:

Passive Metarule

$$\begin{array}{c} \mathrm{VP} \sim [\mathrm{PAS}] \longrightarrow W,\ \mathrm{NP} \\ \Downarrow \\ \mathrm{VP}[\mathrm{PAS}] \longrightarrow W,\ (\mathrm{PP}[by]) \end{array}$$

This assures that it cannot apply to its own output and that the principle is satisfiable. Note that this is not in the class of metarules as formally defined by GKP&S, since these can refer only to categories not their negations. The direct statement of the principle, however is within $L^2_{K,P}$.

Similarly, given the restriction, in GKP&S, to single (long-distance) extractions, we can interpret STM1 as applying only to local trees in which no [+NULL] categories occur.

The restriction to finite closure is a meta-level restriction in that we do not explicitly require it. It is, however, not a stipulation, but rather a consequence of the way in which the metarules are encoded. Metarules that do not exhibit finite closure will be encoded as principles which cannot be satisfied by any definition.[28]

[28] Note that, while it is decidable if a given Φ_G satisfies our interpretation of a given metarule (since the

5.3 Inherited features independent of the form of Φ_G

The fundamental idea of FFP, CAP, and HFC at the theoretic level is that there are specific sets of features on which nodes must agree when they occur in specific configurations, unless they are explicitly required to disagree. In GPSG grammars, the most basic class of explicitly required features are the inherited features which are defined in terms of the ID rules (as expanded by the metarules). In abstracting away from the form of Φ_G we can no longer define inherited features in these terms. This is the key difficulty in stating these principles at the theoretical level—how to capture the exceptions encoded in the ID rules without stipulating the form of Φ_G. The resolution of this issue for inherited features that we develop in this section is sufficient to lift our treatment of FFP, CAP, HFC, and FSDs from the descriptive level to the theoretic level. The idea underlying our approach is the notion, introduced in Section 5.1, that the ID rules of a GPSG grammar roughly correspond to the set of least specified local trees (instances of local contexts) that occur in the set of trees licensed by the grammar.[29]

For $C_i \in \mathbf{C}^T$, let $\langle C_0, \ldots, C_n \rangle$ refer to the local tree in which the category C_0 dominates the categories $C_1 \ldots C_n$. We can identify sets of points forming such a local tree with the formula

$$\langle C_0, \ldots, C_n \rangle (x_0, \ldots, x_n) \equiv \text{Children}(x_0, \ldots, x_n) \wedge \bigwedge_{i \leq n} [C_i(x_i)].$$

We will say a local tree $\langle C_0, \ldots, C_n \rangle$ *subsumes* another $\langle C'_0, \ldots, C'_n \rangle$ (i.e., $\langle C_0, \ldots, C_n \rangle \sqsubseteq \langle C'_0, \ldots, C'_n \rangle$) iff each of the C'_i is an instance of every category the corresponding C_i is an instance of, that is, if the positive component of each C'_i is an extension of the positive component of C_i. This can be expressed as a formula

$$\langle C_0, \ldots, C_n \rangle \sqsubseteq \langle C'_0, \ldots, C'_n \rangle \equiv \bigwedge_{i \leq n} [(\forall x, y)[(C_i(x) \wedge C'_i(y)) \rightarrow \bigwedge_{p \in \mathbf{P}} [P(x) \rightarrow P(y)]]].$$

The idea is that there is a set of minimal local trees that subsume every local tree occurring in a tree in the models of Φ_G. We would like to take these trees as filling the role of the ID rules of a GPSG grammar in determining the set of inherited features. However, these trees will include not only all inherited features, but also all features instantiated by FCRs. We need to consider, then, the distinction between these two sets of features, specifically, why does the FFP propagate features specified by FCRs but not those specified by ID rules and why does the CAP only propagate those specified by the ID rules?[30]

The relevance of FCRs to FFP

We will explore the issue of FCRs only for FFP. While we suspect a similar analysis can be made for CAP, it seems unlikely to add much weight to our argument, and we have not carried it out.

resulting instance of that interpretation is an $L^2_{K,P}$ sentence), it is not necessarily decidable if there is any Φ_G that satisfies that interpretation.

[29] This is a case where the distinction between *local contexts* (defined in terms of $\mathbf{P}$) and *local trees* (defined in terms of $\mathbf{P}^T$) is crucial. The least specified local context occurring in any set of trees, of course, specifies no features.

[30] The HFC does not distinguish these features—all features inherited from ID rules or instantiated by FCRs, the FFP or the CAP override the requirements of HFC.

For FFP the relevant case seems to be the propagation of SLASH features that are introduced as a result of the STM1 metarule. For instance, consider the rule

$$\mathrm{VP} \longrightarrow \mathrm{H}[2],\ \mathrm{NP}$$

STM1 converts this to

$$\mathrm{VP} \longrightarrow \mathrm{H}[2],\ \mathrm{NP}[+\mathrm{NULL}].$$

FCR 19 requires all [+NULL] nodes to be [+SLASH] as well. Since all NULL strings are of form α[+NULL]/α where the 'α's are the same category, lexical insertion will require the NP to be of the category NP[+NULL]/NP. Since the slash is instantiated (by the FCR), it will be propagated to the VP by the FFP. All local trees that are licensed by the rule, then, will be instances of

$$\begin{array}{ll} \mathrm{VP/NP} & \mathrm{H}[2] \\ & \mathrm{NP}[+\mathrm{NULL}]/\mathrm{NP} \end{array}$$

The issue is that, in GPSG, the FFP propagates the SLASH feature because it is instantiated rather than inherited, while, if we take the features of the minimal category instantiating the NP to be inherited then the SLASH will not propagate.

It is not at all obvious, however, why generalizations about feature co-occurrences that are expressed as FCRs should affect the class of features that are governed by FFP/CAP. The distinction seems to motivated more by the needs of the grammar mechanism than by linguistic considerations. Suppose, on the other hand, we were to express STM1 as

$$\begin{array}{c} \mathrm{X} \longrightarrow W,\ \mathrm{X}^2 \\ \Downarrow \\ \mathrm{X/X}^2 \longrightarrow W,\ \mathrm{X}^2[+\mathrm{NULL}]. \end{array}$$

Here again, X^2[+NULL] becomes X^2[+NULL]/X^2 by FCR 19 and agreement between the X^2s on the right hand side of the resulting rule will be enforced by lexical insertion. We must account for the agreement between the SLASH feature on the right hand side with that of the parent. Note that the modified STM1 has a form similar to STM2:

$$\begin{array}{c} \mathrm{X} \longrightarrow W,\ \mathrm{V}^2[+\mathrm{SUBJ},\ \mathrm{FIN}] \\ \Downarrow \\ \mathrm{X/NP} \longrightarrow W,\ \mathrm{V}^2[-\mathrm{SUBJ}]. \end{array}$$

We might pursue this analogy in treating agreement. In the case of instances of rules generated by STM2, agreement is required between the value of the SLASH feature of the X/NP and the AGR feature of the V^2[−SUBJ] because the latter is a predicative category without a controlling sibling. In our interpretation of CAP, agreement is required because the V^2[−SUBJ] will be marked Target and will have no Controller sibling. We might similarly mark the slashed child in instances of rules introduced by STM1 as targets. (Although this would have to be justified within the context of the theoretical basis for control relationships.) Consequently, CAP would apply to these instances as well, enforcing agreement between the SLASH features.

It may be interesting to examine how this approach would treat a more complicated example. Consider the result applying (the modified) STM1 to the ID rule for *believe*:

$$\mathrm{VP/X}^2 \longrightarrow \mathrm{H}[17],\ \mathrm{NP}[+\mathrm{NULL}],\ \mathrm{VP}[\mathrm{INF}].$$

An example would be the VP in

$[_{\mathrm{NP}}$the dog we $[_{\mathrm{VP}}$believe to be vicious]].

Here the infinitival VP complement is an agreement target controlled by the NP[NULL]. Consequently, the value of its AGR feature must mirror the NP on the relevant features. As before, FCR 19 and lexical insertion will require the NP to have a SLASH feature that takes the same values. The approach we are suggesting here would mark the NP as an agreement target as well. Thus it would be both a controller and a target and both it and the VP complement would be targets within the same local tree. Unlike the VP, however, the NP has no controlling (proper) sibling. Consequently, the parent will be taken as the controller and agreement between the value of its SLASH feature and that of the NP (and ultimately the value of AGR feature of the VP complement) will be required by CAP.

Distinguishing marked contexts

A more fundamental difficulty with replacing the notion of ID rules with the notion of the minimal licensed local trees is that there is no guarantee that the marked cases in which, for instance, an FSD may be overridden, is not subsumed by the unmarked case in which it must apply. This is, in fact, the case for every instance of STM1. Consider the pair of ID rules

$$\mathrm{ID}_1: \quad \mathrm{VP} \longrightarrow \mathrm{H}[2],\ \mathrm{NP} \qquad\qquad \mathrm{ID}_2: \quad \mathrm{VP/NP} \longrightarrow \mathrm{H}[2],\ \mathrm{NP}[+\mathrm{NULL}]$$

in which ID_2 is derived from ID_1 by (the modified) STM1. Assuming the other components of the grammar do not interfere, every local tree instantiating ID_2 will be subsumed by an instance of ID_1. In GPSG this is not an issue since the fact that the instance of ID_2 is marked with respect to FSD 2 (which requires $\neg$[+NULL]) is indicated by the presence of the ID rule licensing it. Here we no longer assume that Φ_G will explicitly specify any ID rules (or rather their corresponding local contexts). Nonetheless it must still distinguish such marked contexts from the unmarked contexts. Presumably, exceptional contexts should be distinguished by linguistically motivated configurations, but, rather than trying to identify these configurations, we will simply assume there is a feature [$\mathrm{Unmarked}_f$] which occurs in every category in which f is not exceptionally required. It is important to note that we do not stipulate that Φ_G distinguish unmarked contexts in this way. It should be possible to replace occurrences of [$\mathrm{Unmarked}_f$] with formulae specifying the relevant configuration. That the distinction must be made in some way, however, is a consequence of our revised notion of inherited features—marked configurations can only be distinguished if they are minimal with respect to subsumption; no feature can obtain inherited status from a context that is subsumed by an unmarked context simply because it is impossible to determine which of those contexts licenses a given tree. Note that we associate the presence of the feature with the unmarked case, since, like all features, it may be freely instantiated. The presence of [$\mathrm{Unmarked}_f$] on a category assures that all instances of it will be flagged as being unmarked. On the other hand, the presence of instances of marked contexts which bear the feature [$\mathrm{Unmarked}_f$] by instantiation will be harmless.

Under this assumption, no instance of an unmarked local context will ever subsume an (ordinary) instance of a marked local context since the unmarked configuration will be defined for [$\mathrm{Unmarked}_f$] while the marked configuration will not.

Reinterpreting the notion of inherited features

Suppose, then, we take a feature to be inherited in a given local tree $\langle C_0, \ldots, C_n \rangle$ iff it occurs in all licensed local trees subsuming $\langle C_0, \ldots, C_n \rangle$. This will include features instantiated by FCRs but not those instantiated by the FFP, CAP, HFC, or FSDs—at least if they actually vary in the models of Φ_G. Note that if they don't vary among those models then they actually represent an FCR for one or more of the categories in the local tree. We can define $\mathrm{Inh}_f(x)$, for a feature f as follows:

$\mathrm{Inh}_f(x) \overset{\mathrm{def}}{\Longleftrightarrow}$

- $\langle C_0, \ldots, C_n \rangle$ is *a* local tree in which x occurs as, say, C_i (there will usually be two such local trees), and
- f is a component of C'_i in every $\langle C'_0, \ldots, C'_n \rangle \sqsubseteq \langle C_0, \ldots, C_n \rangle$ that occurs in a model of Φ_G.

We capture this formally by defining first

$$\mathrm{LT}(C_0, \ldots, C_n, y_0, \ldots, y_n) \equiv \mathrm{Children}(y_0, \ldots, y_n) \wedge \bigwedge_{i \leq n} [C_i(y_i)],$$

and

$$\mathrm{LT}_i(C_0, \ldots, C_n, x) \equiv (\exists y_0, \ldots, y_n)[\mathrm{LT}(C_0, \ldots, C_n, y_0, \ldots, y_n) \wedge x \approx y_i].$$

$\mathrm{LT}(C_0, \ldots, C_n, y_0, \ldots, y_n)$ is true whenever $\langle y_0, \ldots, y_n \rangle$ is a local tree that is subsumed by $\langle C_0, \ldots, C_n \rangle$.

$\mathrm{LT}_i(C_0, \ldots, C_n, x)$ is true at x whenever it is the i^{th} point in such a local tree.

Then letting m be the maximum branching factor of the models of Φ_G (suspending again the treatment of infinite schemata) we get

$$\begin{array}{l}
\mathrm{Inh}_f(x) \equiv \\
\quad \bigvee_{n \leq m} \bigvee_{C_0, \ldots, C_n \in \mathbf{C}^T} \bigvee_{i \leq n} [\\
\qquad \mathrm{LT}_i(C_0, \ldots, C_n, x) \rightarrow \\
\qquad \bigwedge_{C'_0, \ldots, C'_n \in \mathbf{C}^T} [\ (\ \langle C'_0, \ldots, C'_n \rangle \sqsubseteq \langle C_0, \ldots, C_n \rangle \wedge \\
\qquad\qquad (\exists \mathbf{P})[\Phi_G(\mathbf{P}) \wedge (\exists y_0, \ldots, y_n)[\mathrm{LT}(C'_0, \ldots, C'_n, y_0, \ldots, y_n)]]) \quad \rightarrow \\
\qquad\qquad (\forall y)[C_i(y) \rightarrow f(y)] \\
\qquad\qquad] \\
\qquad]
\end{array}$$

5.4 Accommodating Unbounded Branching

We return now to the issue of adapting our treatments of metarules and inherited features to the case of sets of trees in which the branching factor is not bounded, that is, sets of trees licensed by infinite GPSG grammars.

Let us assume, for now, that, in the class of definitions we will admit, the only infinite schemata are those in which individual categories are iterated by the Kleene closure.[31] Note

[31] We will suggest momentarily that such a restriction could be assumed as a consequence of a strengthened version of the ECPO property.

that, since one can only distinguish $\mathbf{card}(\mathbf{C}^T)$ categories and since $\mathbf{C}^T$ is finite, there is a finite bound on the number of iterated categories occurring on the right hand side of a rule.

In encoding a local tree in which branching is bounded by m we used used a finite schema

$$\bigwedge_{z \in \vec{z}} [\phi_z(z)]$$

in which $\vec{z}$ ranged over all sequences of m variables and the ϕ_z were drawn from all sequences of formulae in $\mathbf{C}$, indexed by $\vec{z}$. Here we will distinguish two sequences of variables, $\vec{z}$ which will pick out the non-iterated categories of the local tree and $\vec{w}$ which will pick out the iterated categories. As we have just established, the number of distinct categories in each of these classes is bounded, and thus the length of the sequences of variables is bounded. In any instance of a local context, every node will be licensed either by one of the non-iterated categories or by one of the iterated categories. Further, if a local context licenses any local tree in which some iterated categories occur, then it licenses all local trees in which any number of those categories occur (or, perhaps, any positive number). Thus, taking z_0 to pick out the parent, we can use the schema

$$\bigwedge_{z \in \vec{z}} [\phi_z(z)] \wedge (\forall z')[z_0 \triangleleft z' \rightarrow (\bigvee_{z \in \vec{z}} [z' \approx z] \vee \bigvee_{w \in \vec{w}} [\phi_w(z')])]$$

where the $\vec{z}$, $\vec{w}$, ϕ_z, and ϕ_w vary as before.

We would like, finally, to justify the restriction to sets of trees that can be generated by grammars in which only single categories occur in the scope of a Kleene star. To begin with, it is clear that, given a restriction to ID rules that do not specify precedence, the Kleene closure can only be applied to single categories. The reason is that $(a, b)^*$ is simply not a regular expression; any regular expression iterating equal numbers of a and b will have to iterate substrings in which they occur in specific sequences.[32] In abstracting the ECPO from ID/LP format, we ignore the fact that there are LP facts other than simple precedence relations between individual categories. The rule

$$S \longrightarrow (ab)^* + (ba)^*$$

(where, again, + denotes alternation) implies no LP ordering of 'a's and 'b's, but nonetheless implies that they will always occur in adjacent pairs. Perhaps the proper principle, then, would be a strengthening of ECPO to rule out any such LP constraints. We need not explore the precise statement of such a principle here, it suffices to note that it would effectively restrict application of the Kleene closure to single categories.

6 Conclusions

Theories of syntax have their empirical foundations primarily in judgments of the grammaticality of strings. On top of this they build a number of levels of theoretical analysis—lexical

[32] GKP&S point out, citing Shieber, that an ID/LP grammar employing such an expression generates the language $a^n b^n c^n$:

$$S \longrightarrow (a, b, c)^* \qquad a \prec b \prec c$$

categories are posited in order to abstract away from the irrelevant details of the lexicon, a hierarchical structure of phrases and clauses are then built on this basis, and the universal characteristics of human languages are expressed in terms of that structure. Consequently, all approaches to syntax share a common character in that they define sets, or classes of sets, of structures. Beyond this theories may make claims about the specific nature of the human language faculty or about the process of language acquisition, etc., but at this level there is little commonality even in the goals of the varying approaches.

Our claim here is that the crux of what theories of syntax have to say about language lies in the abstract properties of the sets of structures they license. This is the level that is most directly connected to the empirical basis of these theories and it is the level at which it is possible to make meaningful comparisons between the approaches. From this point of view, grammar formalisms (or formal frameworks) are primarily means of presenting these properties. Many of the apparent distinctions between formalisms, then, may well be artifacts of their presentation rather than substantive distinctions between the properties of the structures they license.

The primary strength of the model-theoretic approach we advocate here is the way in which it can cut through these idiosyncrasies of presentation. By treating theories of syntax as presentations of (classes of) sets of ordinary mathematical structures we gain considerable clarity in understanding the claims that a theory makes about the structure of utterances and in understanding the consequences of those claims both in concert and in isolation. We believe that the variety of unresolved issues uncovered by the re-interpretation of GPSG we offer here illustrates the potential this approach has for illuminating existing theories. But beyond that, the fact that the range of approaches to syntax can be understood as a variety of ways of specifying properties of structures and sets of structures suggests that this approach might provide a common framework for comparing these approaches. This, in turn, raises the prospect of reducing distinctions in the mechanisms underlying these approaches to distinctions in the properties of the structures they license. At that point the empirical consequences of these distinctions, if in fact they exist, should become apparent.

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.

[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.