The Nature and Structure of a Computational Linguistic Theory

Carl Pollard

Carnegie-Mellon University

Proceedings of ROCLING I(1988)
R.O.C. Computational Linguistics Workshops I pp 289-318

中華民國第一屆計算語言學研討會論文集 289-318 頁

The Nature and Structure of a Computational Linguistic Theory

Carl J. Pollard
Carnegie Mellon University

1. Introduction

Within the past decade, grammar formalisms based upon the unification of feature structures have come to play a central role in many different research traditions in theoretical and computational linguistics.[1] Within linguistic theory, perhaps the most familiar examples come from generalized phrase structure grammar (GPSG, Gazdar et al., 1985), where feature structures are employed to model syntactic categories, and from lexical-functional grammar (LFG, Bresnan, ed., 1982), where they model a posited level of syntactic representation called f-structure (functional structure) which contains information about the grammatical relations of an expression. Somewhat simplified examples of such feature structures are indicated by the attribute-value matrix (AVM) diagrams in Figs. 1 and 2:

$$\begin{bmatrix} \text{MAJ} & \text{n} \\ \text{BAR} & 2 \\ \text{CASE} & \text{nom} \\ \text{NFORM} & \text{norm} \\ \text{AGR} & \begin{bmatrix} \text{PER} & \text{3rd} \\ \text{NUM} & \text{sng} \\ \text{GEN} & \text{fem} \end{bmatrix} \end{bmatrix}$$

Figure 1: GPSG category for the pronoun "she"

$$\begin{bmatrix} \text{PRED} & \text{'try <SUBJ XCOMP>'} \\ \text{SUBJ} & \boxed{1}\ \text{John} \\ \text{XCOMP} & \begin{bmatrix} \text{PRED} & \text{'leave <SUBJ>'} \\ \text{SUBJ} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

Figure 2: LFG f-structure for "John tried to leave"

But it is important to be aware that feature structures can be used to model a very wide variety of linguistic objects (not just syntactic ones), depending on the subject matter of the theory involved. For example, in the recent trend known as "multi-tiered" phonology,

[1]The research reported herein was supported by a grant from the National Science Foundation (BNS-87-18156).

$$\begin{bmatrix} \text{LARYNGEAL} & \begin{bmatrix}\text{SPREAD} & + \\ \text{CONSTRICTED} & - \\ \text{VOICED} & -\end{bmatrix} \\ \text{SUPRALARYNGEAL} & \begin{bmatrix} \text{MANNER} & \begin{bmatrix}\text{NASAL} & - \\ \text{CONTINUANT} & + \\ \text{STRIDENT} & +\end{bmatrix} \\ \text{PLACE} & \begin{bmatrix}\text{CORONAL} & + \\ \text{ANTERIOR} & + \\ \text{DISTRIBUTED} & -\end{bmatrix} \end{bmatrix}\end{bmatrix}$$

Figure 3: Feature specification for /s/ (after Clements, 1985)

$$\begin{bmatrix} \text{DESCRIBED-SITUATION} & \left\{ \begin{bmatrix} \text{RELATION} & \text{BELIEVE} \\ \text{EXPERIENCER} & \boxed{1} \\ \text{THEME} & \begin{bmatrix}\text{RELATION} & \text{ASLEEP} \\ \text{THEME} & \boxed{2}\end{bmatrix} \end{bmatrix} \right\} \\ \text{BACKGROUND-SITUATION} & \left\{ \begin{bmatrix}\text{RELATION} & \text{NAMED} \\ \text{ARG1} & \boxed{1} \\ \text{ARG2} & \text{WANG}\end{bmatrix}, \begin{bmatrix}\text{RELATION} & \text{NAMED} \\ \text{ARG1} & \boxed{2} \\ \text{ARG2} & \text{GUNJI}\end{bmatrix} \right\} \end{bmatrix}$$

Figure 4: Situation Schema for "Wang believes Gunji is asleep"

the "geometry" of phonological features is analyzed using feature structures such as the one shown in Fig. 3.

And in various situation-based approaches to linguistic semantics (such as the situation schemata of J.-E. Fenstad and his collaborators (1987)), the semantic content and presuppositions of an utterance might be represented roughly as in Fig. 4.

In this paper, my primary goal is to give an overview of the organization and linguistic content of a particular feature-structure based linguistic theory, head-driven phrase structure grammar (HPSG), which I have developed with the help of colleagues at Stanford University and Hewlett-Packard Laboratories since about 1984. At the end, I will also show how the theory can be expressed using a formal logical language, and try to explain in what sense HPSG can be viewed as a *computational* theory.

Among linguistic theories, perhaps the most distinctive characteristic of HPSG is that linguistic objects of all kinds, be they syntactic, phrase-structural, semantic, phonological, or pragmatic, are all modelled by feature structures (though for simplicity I will touch on phonology and pragmatics only tangentially in this discussion). Thus, for example, the English third-person singular verb *walks* is modelled (with some simplifications) as the feature structure depicted by the AVM in Fig. 5, or alternatively by the graph representation

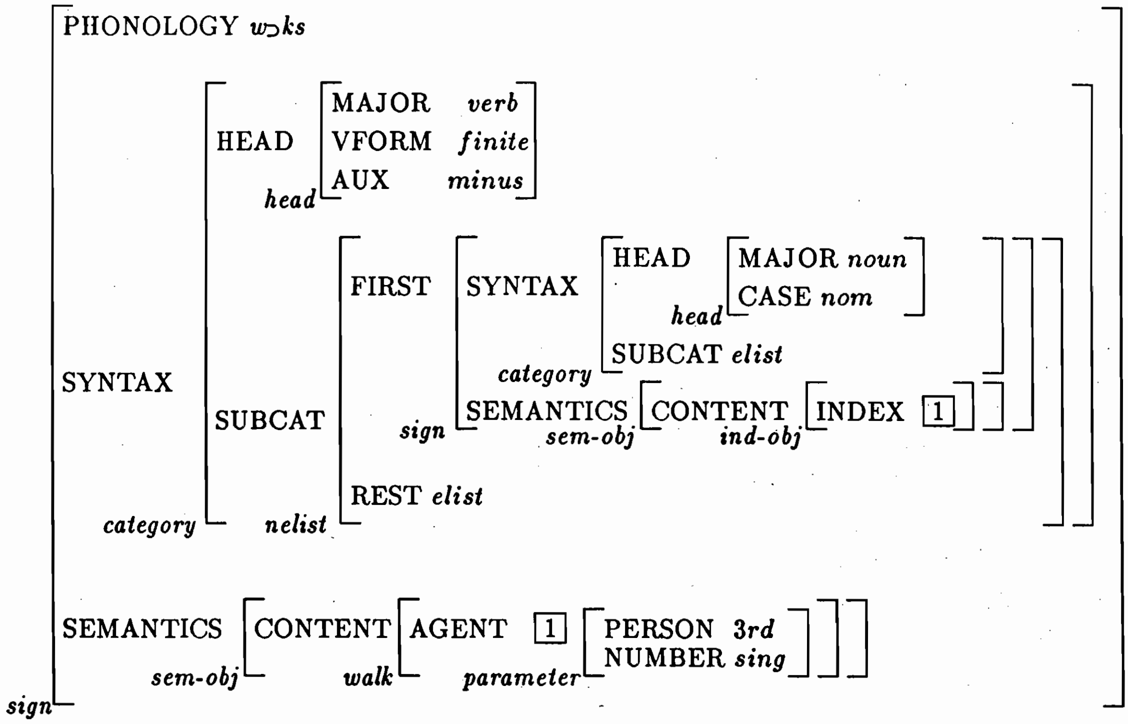

Figure 5: The sign *walks* (AVM notation)

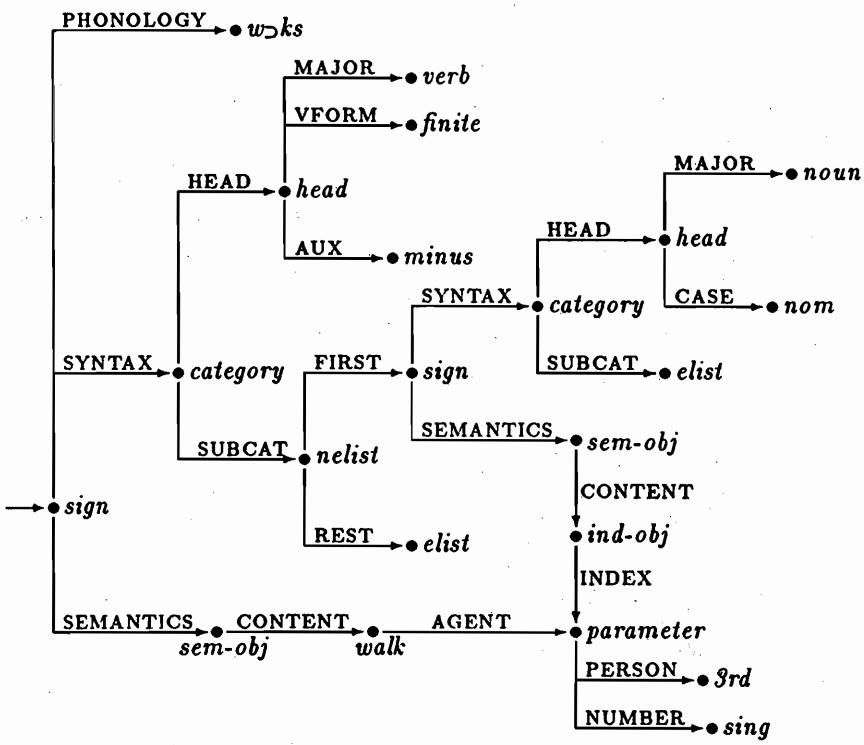

Figure 6: The sign *walks* (graph notation)

in Fig. 6. I will try to explain the significance of objects of this kind, and how they fit into a theory of natural language, in due course. But before becoming enmeshed in the technical details, I will try to give a rough indication of what kind of thing I think a linguistic theory should be. I hope these prefatory remarks will help make a little clearer how the linguistic phenomena being described, the feature structures, and the formal logical theory all relate to each other.

2. The Nature of Linguistic Theory

In a mathematical theory about an empirical domain, the phenomena under study are *modelled* by certain mathematical structures, certain aspects of which are conventionally understood as corresponding to observables of the domain. The theory then talks about, or is *interpreted into*, the model, *not* into the phenomena; the predictive power of the theory arises from the conventional correspondence between the model and the empirical domain. An informal theory is one that talks about the model using natural language, e.g., a technical dialect of English, Chinese, or Japanese. But as theories become more complicated and their empirical consequences less clear, the need for formalization arises. In cases of extreme formality, of course, the theory is cast as a set of axioms in a logical language, and the modelling structures serve as the intended model-theoretic interpretation of expressions in the logic.

For example, in a standard model of celestial mechanics, the positions and velocities of the planets and sun are represented by vectors in a Euclidean space ("phase space"), their masses by positive real numbers, and their motions by certain vectorfields (suitably well-behaved "flows") on the space. The model is not the solar system, but certain aspects of it represent aspects of the solar system of interest to the physicist. Other aspects, such as the size of the planets, interstellar dust, and relativistic effects that become significant only at velocities approaching the speed of light, are not taken into account. In a formal theory based on such a model, the underlying logic is a standard first-order language of set theory and the axioms are certain differential equations which the flows are required to satisfy. An observed motion is then predicted insofar as it agrees—under the conventional correspondence—with an admissible flow, i.e., one that satisfies the equations (or obeys the theory). This situation is illustrated in Fig. 7.

In my view, a linguistic theory should have exactly the same relation to the universe of possible linguistic objects under study as a mathematical theory of celestial mechanics has to the possible motions of the planets (this notwithstanding the unfortunately common belief among syntacticians that linguistic theories cannot be falsified by "mere" facts). In feature-structure-based linguistic theories, the modelling structures of choice, the analogs of the space physicist's flows, are of course feature structures.[2] As I mentioned above,

[2]For an introduction to feature structures as employed in theoretical and computational linguistics, see Shieber, 1986. A somewhat more technically oriented, but still informal, survey is provided by Pereira, 1987. For linguistic motivation, see Fenstad et al., 1987, and Sag et al., 1986.

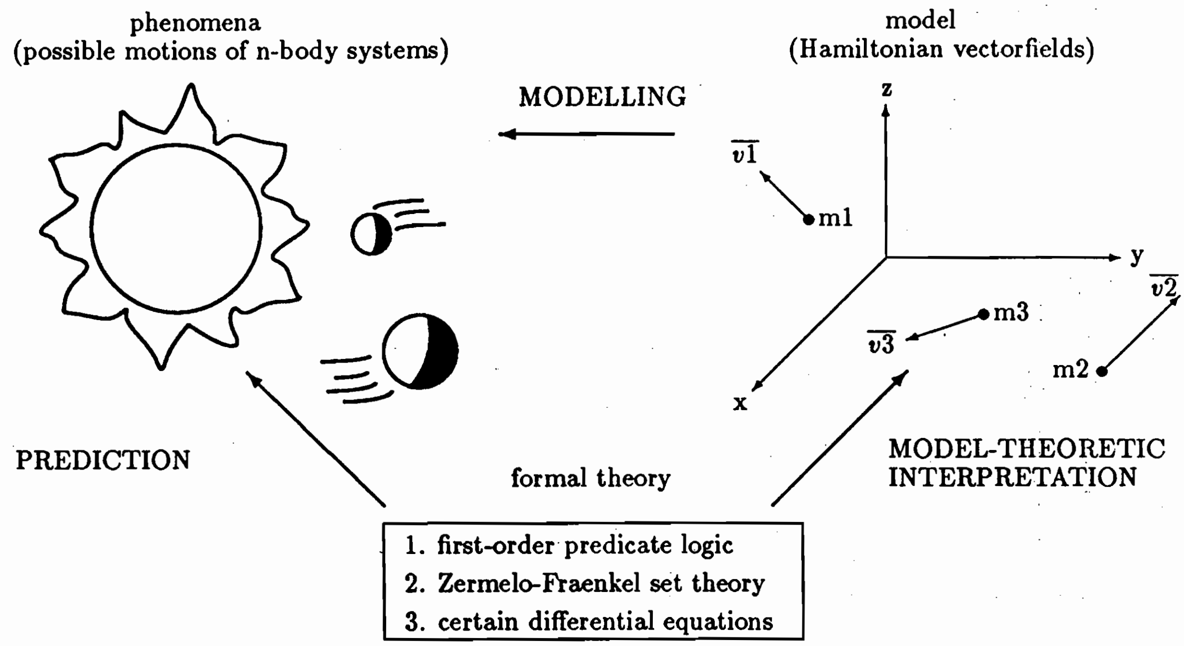

Figure 7: The nature of a physical theory

depending on the research framework, the feature structures are used to model whatever kinds of linguistic entities are posited and considered to be under investigation. The role of the linguistic theory, then, is to give a precise specification of which feature structures are to be considered admissible. The linguistic entities which correspond to the admissible feature structures constitute the predictions of the theory. Just as in other empirical domains, the need has arisen to formalize such theories, and in the past couple of years various languages for specifying constraints on feature structures have been proposed (Rounds & Kasper, 1986; Moshier & Rounds, 1987; Johnson, 1987; Gazdar et al., 1987). Formalisms such as PATR-II (Shieber et al., 1983; Shieber, 1984; Karttunen, 1986) and functional unification grammar (FUG, Kay, 1979 and 1985) specifically intended to facilitate computer implementation of linguistic theories can be viewed in much the same way. Constraints expressed in such such languages can be regarded as the linguist's analog to the space physicist's differential equations. In the case of HPSG, to which I will turn shortly, the overall picture is something like that shown in Figure 8.

The nature of the model and of the formal theory will, I hope, become quite clear in due course, but first a few words are in order about the phenomena, the subject matter of the theory. What are the linguistic objects that HPSG—or for that matter any linguistic theory—is about?

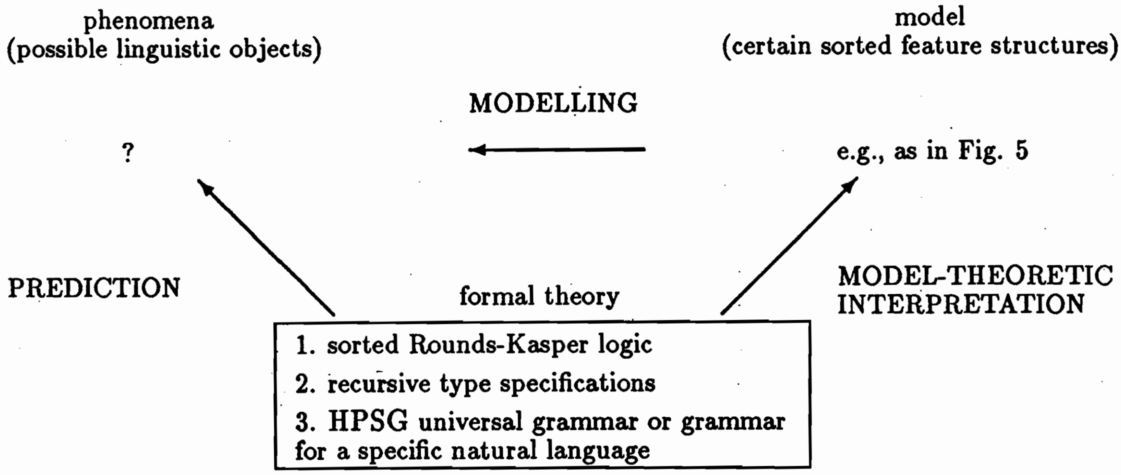

Figure 8: The nature of a linguistic theory

3. Types of Linguistic Objects and Feature Structure Sorts

The central goal of linguistic theory is to characterize what it is that every language user knows by virtue of being a language user, i.e., universal grammar. And a theory of a particular language—a grammar—characterizes what it is that all users of that language know in common. Indeed, from the linguist's point of view, that is what the language is. But what does it consist of? One thing that it certainly does not consist of is individual linguistic events or utterance tokens, for these are not things knowledge of which is shared among the members of a speech community. Instead, what the members of a linguistic community know in common, that makes communication possible, is a system of linguistic types. For instance, the type of the sentence *I'm sleepy* is part of that system, but no individual token utterance of it is.

Just what sorts of things these linguistic types are is another question. In fact, the precise ontological status of linguistic types is the subject of a very long-standing debate among various schools of linguistic conceptualists, such as Ferdinand de Saussure and Noam Chomsky, who take them to be mental objects, and linguistic realists, such as Leonard Bloomfield and Jon Barwise, who consider them to belong to extramental reality. Thus, depending on our philosophical predispositions, we might identify linguistic types with such psychological entities as Saussure's signs, or with certain presumably nonmental objects of situation theory (situation types, or perhaps infons). Fortunately, as Richmond Thomason (in press) has pointed out, a science need not have solved its foundational problems in order to be successful: the interminable philosophical debate over the meaning of quantum mechanics does not seem to have diminished its predictive power. Our immediate concern is the internal architecture of the system that these types form, not with that system's ultimate ontological status.

In the early days of generative grammar (e.g., Chomsky, 1957), the types singled out for attention were the sentences, considered as strings of phonetic shapes. Correspondingly, a grammar was just a computational device, e.g., a context-free grammar or a transformational grammar, that enumerated a set of strings. Current linguistic theory, of course, is more demanding: the linguistic types *par excellence*, the expressions—or, to borrow Saussure's term, the signs—include not only sentences, but also the words and phrases, even discourses. And a sign is considered to consist not just of a phonetic form, such as the PF of government-binding theory (GB, Chomsky, 1981), but of other attributes ("levels of representation") as well. For example, most current linguistic frameworks posit a kind of surface constituent structure, whether it is called s-structure (GB), c-structure (LFG), or phrase structure (GPSG). Some posit an additional attribute that deals with such syntactic notions as subject, object, specifier, and adjunct, e.g., GB's d-structure or LFG's f-structure. And some frameworks further require an attribute of expressions that deals with various aspects of the things the expression describes, such as predicate-argument structure and quantifier scope. Examples of this are the semantic interpretations of GPSG (encoded by formulas of Montague's intensional logic), and GB theory's LF.

In HPSG, all four of these attributes of expressions are assumed to exist. Thus all signs have a PHONOLOGY, a SYNTAX, and a SEMANTICS, as illustrated by the feature structure for the lexical sign *walk* in Fig. 5. In addition, all phrasal signs (i.e., those which are not lexical) have a DAUGHTERS attribute which specifies the immediate constituent structure (the "surface structure" or "local tree") of the sign. Notice that in an HPSG constituent structure, not only is each daughter of the sign in question indicated, but also the way in which it is a daughter (e.g., head daughter, complement daughter, adjunct daughter, etc.). This is illustrated by the partial sketch in Fig. 9 of the feature structure for the phrasal sign *Bill sneezed.* The same thing is shown in Fig. 10 using a somewhat more familiar abbreviatory notation for PHONOLOGY and DAUGHTERS which I hope will be self-explanatory. I will return to the topic of constituent structure shortly.

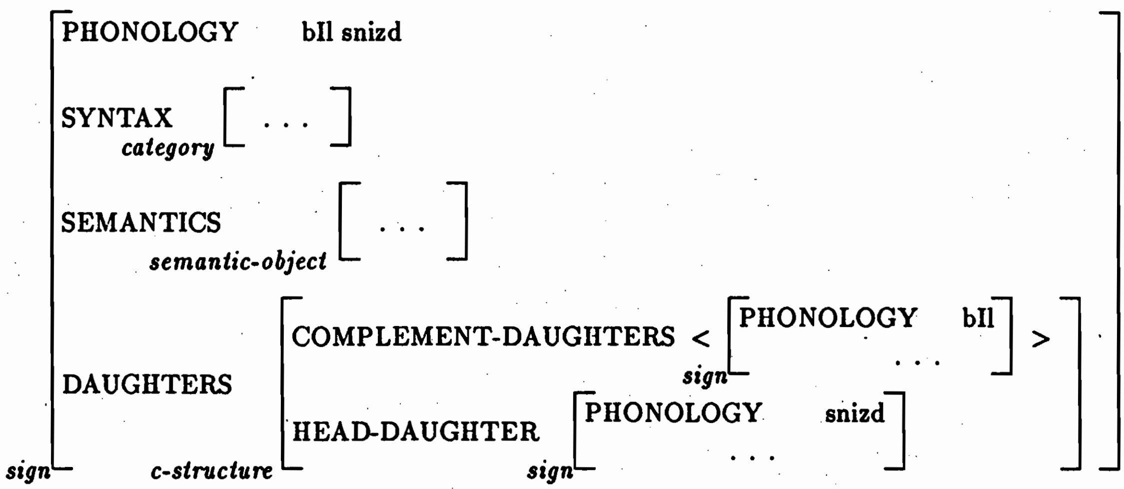

Figure 9: Modelling of constituency by feature structures

As in many current linguistic frameworks, most of the action in HPSG is in the lexicon, so as good a way as any to introduce the framework is to consider the feature structure shown

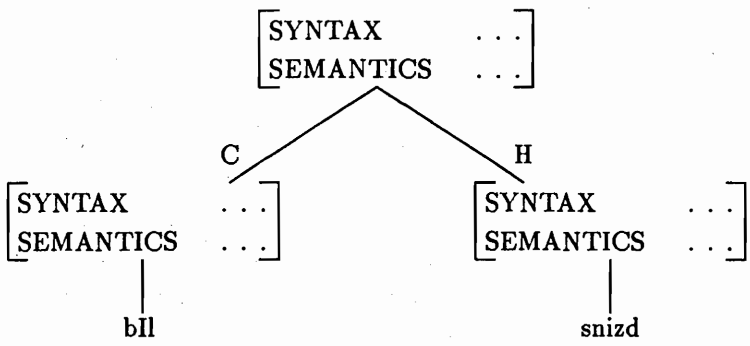

Figure 10: Tree-like notational shorthand for PHONOLOGY and DAUGHTERS

in Figure 5 in some detail. In this discussion I will assume a working familiarity with the basic workings of feature structures, including such notions as subsumption, generalization, disjunction, unification, and structure-sharing (token identity of substructures). For background on such matters, I strongly recommend the informal overviews by Shieber, 1986, and Pereira, 1987, mentioned above.

Mathematically, feature structures can be conceived in a number of ways, e.g., as finite state machines of a certain kind, or as elements of certain partially ordered sets with considerable algebraic and topological structure (Scott domains). For present purposes, though, it is most convenient to think of a feature structure as a rooted directed graph, with a label (the name of an attribute) on each arc. In HPSG we also adopt the practice of letting each node in the graph have a symbol, called a *sort*, assigned to it.[3] Intuitively, the sort tells what basic type of object from the empirical domain is being modelled by the feature structure. In the present case, of course, the sort is *sign*.

Now in general it is assumed that different types of objects in the empirical domain have different attributes. For example, among the attributes of a sign are its phonology, syntax, and semantics (and its daughters too, if the sign is a phrase). This is reflected in the model by the fact that what arc labels a feature structure has depends on the sort symbol assigned to its root; thus a feature structure of sort *sign* has arcs out of the root node labelled PHONOLOGY, SYNTAX, and SEMANTICS. But the objects studied in HPSG are not only signs, but a wide range of other types of linguistic objects as well. Among these are those types of objects which can occur as values of the various attributes of signs, such as phonological shapes, (syntactic) categories, semantic objects, and constituent structures, as well as the types of objects which occur in turn as values of *their* attributes, and so on until we bottom out at types (such as those which occur as values of attributes like CASE, PERSON, NUMBER, VFORM, etc.) which have no attributes of their own.

[3]This practice is borrowed from the field of knowledge representation (e.g., Ait-Kaci, 1984), where the feature structures are usually called *frames*, and the sorts are the names of "generic" frames. For a mathematical exposition of feature structures with sorts, see Pollard, 1988a.

4. Head Features, Subcategorization, and the Obliqueness Order

Let us now consider the SYNTAX and SEMANTICS attributes in some detail.[4] The SYNTAX attribute of a sign is a category, which in turn has two attributes of its own, the HEAD and the SUBCATEGORIZATION.[5] The value of HEAD is an object which contains specifications for such features as MAJOR (roughly, part of speech), CASE, and VFORM (verb inflectional form). As we will see shortly, a principle of universal grammar called the *Head Feature Principle* (HFP) ensures that the HEAD features of a word are always shared with their phrasal projections. For example, if a noun has nominative case then so does the noun phrase which it heads; similarly, if a verb is finite, then so is any verb phrase or sentence headed by that verb.[6]

The other attribute of a sign's category, the SUBCATEGORIZATION (SUBCAT), is a list of the various dependent phrases that the sign characteristically combines with in order to become "saturated" or "complete", such as subjects, objects, specifiers, verbal and sentential complements, and so forth.[7] Thus the SUBCAT value plays much the same role in HPSG as d-structure in GB or f-structure in LFG, i.e., to specify grammatical relations of phrases as they are "projected" from lexical entries. Note that HPSG's list notation is similar to that employed in the LISP programming language, i.e., a nonempty list (*nelist*) of signs is represented as an object whose FIRST is a sign and whose REST is a list of signs. In the present case the list has length one so the REST is the empty list (*elist*, like LISP's "NIL"). Using angle brackets as an alternative notation for lists, the subcategorization of *walks* can be expressed more concisely as in Fig. 11a.

For further succinctness in Fig. 11, I have employed standard abbreviations for certain feature structures, such as those shown in Fig. 12. Note in particular that the *indices* of NPs (of which more below) are indicated by subscripts. Other abbreviations for feature specifications (such as those for CASE and VFORM in Fig. 13) should be self-explanatory.

Thus we see that the intransitive verb *walks* subcategorizes for a third-singular nominative NP, namely its subject, while transitive *sees* requires an additional accusative NP object, and ditransitive *gives* requires yet another accusative NP, its second object. In the case of *tries* an infinitival VP complement is required in addition to the subject, while *persuades* and *promises* call for a subject, an object, and an infinitival VP complement. Notice that subject-verb agreement and case assignment are both treated as aspects of subcategorization,

[4]For expository simplicity, I am indulging in a systematic abuse of language here, wherein a linguistic object is identified with the feature structure that models it.

[5]This is a simplification. I am ignoring here an additional distinction within SYNTAX, wherein HEAD and SUBCATEGORIZATION are treated as LOCAL attributes, as opposed to NONLOCAL (or BINDING) attributes such as SLASH. The NONLOCAL attributes figure in the analysis of long-distance dependency phenomena such as "wh-movement"; these will be treated at length in Pollard (in preparation).

[6]HPSG follows GPSG here in assuming that the head of a sentence is a (possibly auxiliary) verb; unlike GB, there is no INFL.

[7]Hereafter, names of sorts and attributes will be abbreviated, for notational ease.

a. walks: < NP[*nom*] >
[*3rd,sng*]

b. sees: < NP[*acc*], NP[*nom*] >
[*3rd,sng*]

c. gives: < NP[*acc*], NP[*acc*], NP[*nom*] >
[*3rd,sng*]

d. tries: < VP[*inf*, SUBCAT < NP $\boxed{1}$ >], NP[*nom*] $\boxed{1}$ [*3rd,sng*] >

e. persuades: < VP[*inf*, SUBCAT < NP $\boxed{1}$ >], NP[*acc*] $\boxed{1}$, NP[*nom*] [*3rd,sng*] >

f. promises: < VP[*inf*, SUBCAT < NP $\boxed{1}$ >], NP[*acc*], NP[*nom*] $\boxed{1}$ [*3rd,sng*] >

g. says: < S[fin], NP[*nom*] >
[*3rd,sng*]

Figure 11: SUBCATEGORIZATION values of some English verbs

$$\underset{X}{NP} = \begin{bmatrix} \text{SYNTAX} & \begin{bmatrix} \text{HEAD} & [\text{MAJ } n] \\ \text{SUBCAT} & < > \end{bmatrix} \\ \text{SEMANTICS} & [\text{CONTENT [INDEX X]}] \end{bmatrix}$$

$$VP = \begin{bmatrix} \text{SYNTAX} & \begin{bmatrix} \text{HEAD} & [\text{MAJ } v] \\ \text{SUBCAT} & < NP > \end{bmatrix} \end{bmatrix} \qquad S = \begin{bmatrix} \text{SYNTAX} & \begin{bmatrix} \text{HEAD} & [\text{MAJ } v] \\ \text{SUBCAT} & < NP > \end{bmatrix} \end{bmatrix}$$

Figure 12: Abbreviations for some frequently used feature structures

rather than by independent subsystems of the grammar. Thus, a third-singular verb is just a verb which subcategorizes for a third-singular subject.

In this connection, it should be noted that the agreement features PERSON, NUMBER, and GENDER, are treated as attributes of NP semantic indices (analogous to logical variables, but actually more akin to the "parameters introduced by uses of NPs" employed in situation semantics), not as syntactic features. Thus agreement is observed between two signs when (1) both make reference to one and the same index, and (2) variation in the value of some agreement attribute is correlated with a difference in shape for at least one of the signs in question. This theory of agreement, which is set forth in detail in Pollard and Sag, 1988, accounts for a wide range of facts that a purely syntactic theory cannot explain, including: the agreement of a pronoun with its antecedent (though they may differ in case); agreement of a controller with the understood subject that it controls; gender agreement between deictic pronouns and their referents (in natural gender languages); the fact that agreement (unlike a syntactic feature such as case) need not be shared between a coordinate NP and its conjuncts; and numerous other phenomena.

It is important to be aware that the order in which elements appear in the SUBCAT list does not correspond directly to "surface" order (i.e., the linear order in which their phonological realizations occur), but rather to the traditional notion of obliqueness, a kind of "deep" ordering of grammatical relations. The convention used here is that more oblique elements appear earlier (further left) on the list, so that subjects are least oblique, followed by objects, then second objects, then other dependent elements. The obliqueness ordering is reflected by constructs in many different syntactic theories, including the accessibility hierarchy of Keenan and Comrie (1977), the 1-2-3-Oblique ordering of terms in relational grammar, and the default ordering of grammatical functions SUBJ-OBJ-OB2 employed in the LFG account of control.

As it happens, in relatively fixed-constituent order languages, we often find that more oblique elements occur later in surface order; e.g., in English the subject precedes the verb (in canonical sentences) or immediately follows it (in inverted sentences), with the object next in line, followed by the second object. This is not the case, however, in languages like Albanian, Hungarian, or Japanese, where grammatical relations are marked primarily by adpositions or case inflection instead of by position. Nevertheless, the obliqueness order is cross-linguistically significant; indeed, it provides accounts of much the same phenomena that GB theory attempts to explain in terms of configurational notions such as c-command, which have no theoretical importance in HPSG. Obliqueness order figures prominently in HPSG accounts of such phenomena as obligatory control of understood subjects (Sag & Pollard, ms.); constraints on "multiple extraction", such as the licensing of parasitic gaps and prohibitions on crossed long- distance dependencies (Pollard, in preparation); and constraints on pronoun binding (Pollard, 1988b and 1989). I will return to the last-mentioned of these presently.

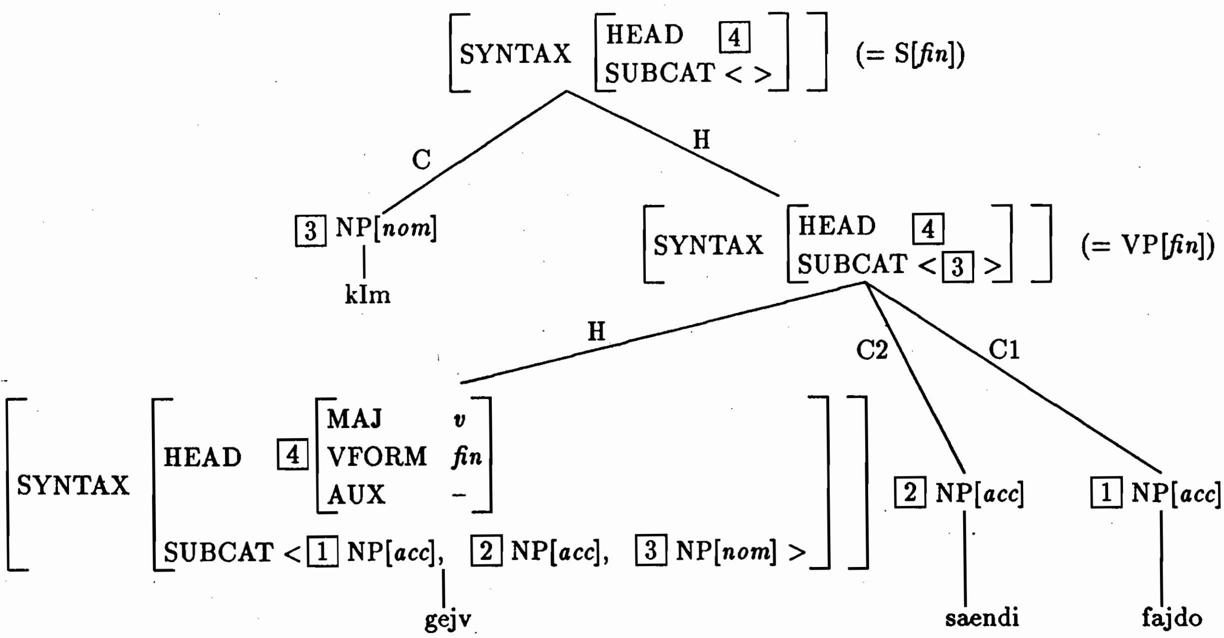

Figure 13: HPSG feature structure for "Kim gave Sandy Fido"

5. Two Principles of Universal Grammar

The SUBCATEGORIZATION attribute is defined not only for words, but for phrasal signs as well. In general the SUBCATEGORIZATION of a sign consists of all the requirements of that sign for dependent elements to combine with that have not yet been satisfied. This is illustrated in Fig. 13. For notational convenience, the SEMANTICS attributes and the sort symbols are omitted.

The things to pay attention to here are the values for the SUBCAT attribute. The lexical head, the verb *gave*, requires a second object ([1]), an object ([2]), and a subject ([3]). Moving up the tree, we come to the first phrasal projection of the verb, the VP *gave Sandy Fido*. In this VP, the requirements for the object and the second object have already been satisfied by the NPs *Sandy* and *Fido*, so only the subject requirement [3] remains to be satisfied. Finally, let us consider the S node (recall that HPSG treats S as a projection of VP). Here the subject requirement has been satisfied by the NP *Kim*; since there are no requirements left to satisfy, the SUBCAT value is the empty list < >.

The general principle being illustrated here is that, in any phrasal sign, the subcategorization requirements that remain to be satisfied are just the subcategorization requirements of the sign's head daughter minus those requirements that were satisfied by the sign's complement daughters. Moreover, it is assumed that the more oblique requirements are satisfied "earlier" (i.e., lower in the tree, not in any temporal sense; there is no procedural assumption about the order in which different parts of the structure are computed). This Subcategorization Principle, which is considered within HPSG theory to be one of the principles of universal grammar, is given a more precise statement in (14a):

(14) Two principles of universal grammar

a. Subcategorization Principle. In any phrasal sign, the SUBCAT list of the head daughter is the concatenation of the list of complement daughters and the SUBCAT list of the mother.

b. Head Feature Principle. In any phrasal sign, the HEAD value is token-identical with the HEAD value of the head daughter.

The analysis given in Fig. 13 also illustrates another principle of universal grammar, the HFP referred to earlier. The point to note in this connection is that the verb *gave* and its two phrasal projections (the VP and the S) all have the same value for the syntactic attribute HEAD, i.e., they have the same specifications for all head features. The HFP is stated in (14b). Below I will introduce a logic that can be used to express such principles more formally.

6. Some Basic Semantic Notions

Let us now complete our guided tour through the feature structure shown in Fig. 5, turning now to the SEMANTICS attribute. The value of this attribute is a model of the meaning of the sign, which we take to include not only the semantic content (a little like literal interpretation, but see below), which is handled within the CONTENT attribute of the SEMANTICS value, but also certain other context-dependent aspects of meaning (roughly speaking, presuppositions and conventional implicatures) which are handled by the CONTEXT attribute. For simplicity I'll consider the CONTEXT attribute only in passing. The overall approach to semantics in HPSG is strongly influenced by situation semantics (Barwise and Perry, 1983), which analyzes reality in terms of such situation-theoretic objects as individuals, relations, and states-of-affairs (now usually called soas or infons), which consist of a relation together with an assignment of objects to the relation's roles (like argument positions). In general, the interpretation of an utterance token will be some such object; for example, the interpretation of a declarative sentence token will be a soa, and the interpretation of a proper name token will be an individual.

However, as linguists we are concerned not with tokens but with types. Hence our semantic contents will not be full-fledged situation-theoretic objects, but rather *parametric* objects, theoretical entities which are made up in part of *parameters* (a little like logical variables) and which yield full-fledged situation-theoretic objects only when their parameters are *anchored* to real objects in a particular utterance context. For example, the interpretation of a particular utterance of the name *Bill* referring to William S.-Y. Wang is just the individual Wang Shiyuan. But as shown in Fig. 15, the semantic content of the sign *Bill* is a object which contains a parameter (as its INDEX attribute). (It also contains a specification for the attribute REFERENCE-TYPE, with possible values *ana* (anaphor), *pro* (pronoun), and *nonpro*

$$\left[\begin{array}{ll}\text{CONTENT} & \left[\begin{array}{ll}\text{REF-TYPE} & \textit{nonpro}\\ \text{INDEX} & \boxed{1}\end{array}\right]_{\textit{ind-obj}}\\ \text{CONTEXT} & \left\{ \left[\begin{array}{ll}\text{ARG1} & \boxed{1}\\ \text{ARG2} & \text{bIl}\end{array}\right]_{\textit{name}} \right\}\end{array}\right]_{\textit{sem-obj}}$$

Figure 15: SEMANTICS value of the NP "Bill"

$$\left[\begin{array}{ll}\text{CONTENT} & \left[\begin{array}{ll}\text{AGENT} & \boxed{1}\end{array}\right]_{\textit{sneeze}}\\ \text{CONTEXT} & \left\{ \left[\begin{array}{ll}\text{ARG1} & \boxed{1}\\ \text{ARG2} & \text{bIl}\end{array}\right]_{\textit{name}} \right\}\end{array}\right]_{\textit{sem-obj}}$$

Figure 16: SEMANTICS value of the sentential sign "Bill sneezed"

(non-pronoun); this will be touched on in the discussion of binding below.) The sign *Bill* also carries the presupposition that in any particular token use, this parameter can only be anchored to an individual named "Bill"; this is indicated within the CONTEXT attribute, whose value is a set of states-of-affairs (which must hold in any appropriate utterance context).

To give another example, a token utterance of the sentence *Bill sneezed* might have as its interpretation the soa <<sneeze, agent: Wang Shiyuan>>, but the semantic content of the sentential sign *Bill sneezed* is the parametrized soa <<sneeze, agent: x>>, where x is a parameter that can only be anchored to individuals named "Bill". In HPSG this is modelled by the feature structure in Fig. 16.

In the case of the verb *walks* (returning to Fig. 5), we see that the SEMANTICS:CONTENT value is just the parametrized soa in Fig. 17.

$$\left[\begin{array}{ll}\text{AGENT} & \left[\begin{array}{ll}\text{PERSON} & \text{3rd}\\ \text{NUMBER} & \text{sng}\end{array}\right]_{\textit{parameter}}\end{array}\right]_{\textit{walk}}$$

Figure 17: SEMANTICS:CONTENT value of the verb "walks"

Of course in a sentence headed by this verb, there will be a further semantic contribution from the subject NP, which will either impose some presupposition upon the agent parameter (if the subject is a proper name) or else "absorb" the parameter by quantifying over it (if the subject is a quantificational NP). The details of how the semantic contributions of different parts of the phrase come together are contained in another principle of universal grammar, the Semantics Principle, which I will not go into here (but see Pollard and Sag, 1987).

$$\begin{bmatrix} \text{SYNTAX} & \begin{bmatrix}\text{SUBCAT} & \langle\ \text{NP}[acc]_{\boxed{3}},\ \text{NP}[acc]_{\boxed{2}},\ \text{NP}[nom]_{\boxed{1}}\ \rangle\end{bmatrix} \\ \text{SEMANTICS} & \begin{bmatrix}\text{CONTENT} & {}_{give}\begin{bmatrix}\text{AGENT} & \boxed{1}\\ \text{GOAL} & \boxed{2} \\ \text{THEME} & \boxed{3}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

Figure 18: Semantic role assignment in the verb "gave"

It is important to note the two occurrences of the parameter $\boxed{1}$ in Fig. 5, one in the SEMANTICS:CONTENT of the sign itself and the other as the value of the INDEX attribute inside the SEMANTICS:CONTENT of the element on the sign's SUBCAT list which corresponds to the verb's subject. This identity of parameters in the lexical entry is what sets up the assignment of the AGENT role to the subject. To give another example, the (partial) feature structure describing the subcategorization and semantic content of the verb *gave* in Fig. 18 shows how the roles AGENT, GOAL, and THEME are assigned to the subject, object, and second object respectively. (Remember that the subscripted boxed numerals tag the parameter in an NP, not the whole NP!) This is analogous to what is called "theta-role assignment under subcategorization" in GB theory, but there are some important differences. For one thing, there is no distinction between "internal" and "external" arguments; the role of the subject is directly assigned by the verb, just as for the objects. Another difference is that HPSG is totally explicit about what "theta-roles" and "NP indices" are: specific components of semantic content (namely argument roles in states-of-affairs and referential parameters of NPs, respectively). In GB theory, both of these centrally important notions remain essentially mysterious, though they are generally assumed to be syntactic in nature.

7. Obligatory Control

The connection between subcategorization and role assignment figures prominently in the HPSG account of obligatory control of understood subjects (e.g., of infinitival VP complements). For example, the well-known distinction between equi and raising control is captured by the (partial) lexical entries for *tried* (equi) and *seemed* (raising) in Fig. 19. (Beware of the notational shorthand here: in both lexical entries the tag $\boxed{2}$ indicates the semantic content of the VP[*inf*] complement.)

There are two essential differences to note. First, *seem* does not assign a semantic role to its subject, while *try* does. This ensures that sentences headed by these verbs have semantic contents like those in Fig. 20 (here the tag $\boxed{1}$ indicates the parameter corresponding to the subject John); the distinction between *persuade* (*Kim persuaded Sandy to be optimistic*) and *believe* (*Kim believed Sandy to be optimistic*) is analogous, except that it hinges on whether or not the *object* is assigned a role. The second difference is that in equi control, only the

$$\begin{bmatrix} \text{SYNTAX} & \begin{bmatrix} \text{SUBCAT} & \langle\, \underset{\text{SEM:CONT}\ \boxed{2}}{\text{VP}[\textit{inf}, \text{SUBCAT}} \langle \underset{\boxed{1}}{\text{NP}} \rangle],\ \underset{\boxed{1}}{\text{NP}[\textit{nom}]} \,\rangle \end{bmatrix} \\ \text{SEMANTICS} & \begin{bmatrix} \text{CONTENT} & {}_{\textit{try}}\begin{bmatrix} \text{AGENT} & \boxed{1} \\ \text{THEME} & \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \text{SYNTAX} & \begin{bmatrix} \text{SUBCAT} & \langle\, \underset{\text{SEM:CONT}\ \boxed{2}}{\text{VP}[\textit{inf}, \text{SUBCAT}} \langle \boxed{3}\, \underset{\boxed{1}}{\text{NP}} \rangle],\ \boxed{3}\, \underset{\boxed{1}}{\text{NP}[\textit{nom}]} \,\rangle \end{bmatrix} \\ \text{SEMANTICS} & \begin{bmatrix} \text{CONTENT} & {}_{\textit{seem}}\begin{bmatrix} \text{THEME} & \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 19: Semantic role assignment in the verbs "tried" and "seemed"

$${}_{\textit{try}}\begin{bmatrix} \text{AGENT} & \boxed{1} \\ \text{THEME} & {}_{\textit{fly}}\begin{bmatrix} \text{AGENT} & \boxed{1} \end{bmatrix} \end{bmatrix} \qquad {}_{\textit{seem}}\begin{bmatrix} \text{THEME} & {}_{\textit{fly}}\begin{bmatrix} \text{AGENT} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

Figure 20: Semantic contents of "John tried to fly" and "John seemed to fly"

indices of the controller and the complement subject are unified (structure-shared), while in raising control, the entire signs (including both syntax semantic content) are unified. This predicts the possibility of syntactic dependencies in raising constructions (such as "quirky" case in Icelandic), but not in equi constructions. For a much more detailed account of control, see Sag and Pollard (ms.).

This account of control differs from the LFG account, where it is assumed that the entire f-structure of the controlling NP is identified with that of the understood subject, in both equi and raising. But in numerous other respects, the control theories of HPSG, GPSG, and LFG differ in common from the GB account, which posits a phonetically null constituent ("PRO") as subject of the complement for the equi verbs and a different phonetically null constituent (a trace left by "NP movement") as subject of the complement for subject-controlled raising verbs.[8] For one thing, as illustrated in Fig. 19, HPSG denies that every subcategorized element is necessarily assigned a role. Another difference is that not every VP has to have a surface subject: controlled VP complements are just VPs (though they have subjects at the "level" of subcategorization). And third, raising to subject and raising to object are treated in essentially the same way; in GB, by contrast, only raising to subject is handled by NP movement, while raising to object is handled by the ancillary device of "S-bar deletion".

[8]The only other use of "NP trace" in GB, viz. in the analysis of passive, is also eliminated in HPSG, which follows LFG in treating passive via lexical rule. Hence HPSG has neither PRO nor NP trace.

Incidentally, there is an analogous dissimilarity between the HPSG and GB accounts of raising to subject and their accounts of long-distance dependency (filler-gap, or "wh-movement") phenomena, such as English-style topicalization, relativization, and wh-questions. In GB theory, at least as it is usually presented, one assumes that the filler actually moves from a lower d-structure position to a higher s-structure position, leaving behind a trace ("variable"). In HPSG, by contrast, the relationship between the filler and the trace is one of partial identity, i.e., structure sharing of semantic content and local syntactic features (HEAD and SUBCAT). These dissimilarities between the GB and HPSG accounts of such phenomena as raising to subject and long-distance dependencies are indicative of a fundamental division between derivational theories, which relate information at different "levels" via procedural and computationally unmanageable destructive operations on structures (such as move-alpha), and unification-based theories, which relate substructures of different attributes via declarative conditions of identity (unification, or structure sharing). This very basic difference is responsible in large part for the seemingly paradoxical situation that transformational grammar, though still considered by many linguists to be the dominant syntactic research paradigm, has as yet exerted only a marginal influence in computational linguistics. Unification-based theories, by contrast, are essentially computational in nature; I will return to this point below.

8. Binding of Anaphors

The job of binding theory is to characterize the linguistic constraints that determine when two NPs in the same sentence must or must not be co-indexed (in HPSG, two signs being co-indexed just means that their values of the INDEX attribute in semantic content are shared). In the standard GB binding theory (Chomsky, 1986), such questions are dealt with entirely in terms of tree configuration. In fact, NP1 is defined to A-bind a co-indexed NP2 just in case (i) NP1 is in the configurational relation of c-command with NP2; and (ii) NP1 is in an argument ("A") position, i.e., it is assigned a theta-role and is either a subject or a complement, where these latter two notions are also defined configurationally. For example, one GB binding principle, usually called condition A, is supposed to characterize the environments in which an anaphor must be bound. Condition A is a constraint on possible indexings I of the NPs in a sentence, as stated in (21). Here the term "anaphor" refers to a class of pronouns that include English-style reflexive and reciprocal pronouns (but not Chinese- or Japanese-style reflexives). This condition looks very complicated, but it is even more complicated than it looks. For example, the theoretical element AGR of INFL must be allowed to count as a possible binder, and in any indexing, the AGR is always co-indexed with the subject of the sentence; also indexings are prohibited from satisfying an additional configurational condition (the "i-within-i" condition) whose precise formulation is unclear. In spite of this complexity, condition A wrongly rules ungrammatical a wide range of acceptable sentences, such as those in (22).

(21) Condition A of the GB binding theory

Suppose X is an anaphor, and let B be the least maximal projection such that:

i. B contains X;

ii. B contains a lexical governor of X;

iii. B contains a subject; and

iv. there is an indexing J of the NPs (and the AGR of INFL, if any) in B such that:

  a. X is A-bound in B under the indexing J; and

  b. J does not result in any NP becoming co-indexed with another NP that dominates it.

Then X must be A-bound in B under the indexing I.

(22)

a. Iran agreed with Iraq that each other's shipping rights must be respected.

b. The two presidential candidates charged that General Noriega had secretly contributed to each other's campaigns.

c. John thought that tiny gilt-framed portraits of each other would make ideal gifts for the twin girls.

d. Which picture of herself did John say Mary believed Bill liked?

There are similar complications and empirical problems with the other conditions of the GB binding theory, which suggest that the configurational approach is on the wrong track.

In HPSG, the GB binding conditions are replaced by analogous conditions that are formulated in terms of obliqueness, not configuration. For example, the HPSG analog of Condition A is stated in 23:

(23) Condition A of the HPSG binding theory

> An anaphor (i.e., [REF-TYPE *ana*]) must be co-indexed with some referential (i.e., non-dummy) less oblique dependent of the same head, provided such exists.

This condition makes the same correct predictions as the GB condition, but it also correctly rules sentences like those in (21) to be grammatical. In addition, it is conceptually much clearer. It is also much easier to check computationally, since nothing more is involved than checking the relative positions of two items on a list (rather than highly complex configurational property which must take all possible alternative indexings into account). The HPSG binding theory is presented in full in Pollard, 1988b, and extended to treat languages with "long-distance anaphors" (such as Chinese and Japanese) in Pollard, 1989.

9. Phrase Structure Rules

Up to this point, most of my discussion has focused upon lexical entries and universal principles, but I have said nothing about the the place of phrase structure rules in the theory. Yet some account is needed of what constitutes a well-formed phrase, beyond universal principles such as the ones I have mentioned. Unlike LFG and GPSG, which employ large numbers of phrase structure rules (or "immediate dominance rules") to characterize the possible phrases, in HPSG it is assumed that there is a very small number of universally available schematic rules (phrase templates, or construction types) from which each language makes a selection. Such schematic rules are the HPSG analogs of the X-bar schemata employed in GB theory, such as those in (24):

(24) Two schematic rules of GB's X-bar theory

a. X″ → Y″ (specifier) X′

b. X′ → X Y″ (complement)

In both theories, the schemata only determine possibilities of immediate domination, not relative order of constituents (which is assumed to be governed by other language-specific constraints). And in both theories, the reason that there are so few phrase-structural schemata is that most of the work in the grammar is being done by the lexical entries.

There is a basic difference between the two theories, however. In GB, grammatical relations are defined configurationally in terms of the X-bar schemata. Thus the Y″ in (24a) is defined to be the specifier, and this is called the subject if X is INFL; the Y″ in (24b) is defined to be the complement, and this is called the object if X is V or P. Indeed, much of the point of Chomsky, 1987 is to expand the territory of such schemata to include the cases where X is INFL or a complementizer. From the point of view of HPSG, such efforts are misguided, since grammatical relations are assumed to be determined by the obliqueness ordering in the subcategorization lists of lexical heads, not by the configurational positions in which they are realized. (On this point, HPSG is very close to LFG, although the latter differs from HPSG by using a keyword encoding rather than a list encoding of grammatical relations.)

In HPSG, the rule schemata do not define grammatical relations; instead they determine their possible configurational realizations in constituent structure (the DAUGHTERS attribute). Three such schemata for the realization of heads and their subcategorized dependents (which in HPSG include subjects) are presented informally in (25); a few additional schemata are needed to handle other constructions, such as filler-gap or topic-comment constructions, coordination, adjunction, and marker-head constructions (e.g., for attaching complementizers or semantically empty adpositions).

(25) Three HPSG universal phrase schemata for heads and complements

a. a saturated (i.e., [SUBCAT <>]) sign with lexical head daughter

b. a sign with lexical head daughter and which subcategorizes only for a subject (i.e., [SUBCAT < X >])

c. a saturated sign with a phrasal head daughter and exactly one complement daughter.

Later I will show how to express such schemata in a formal logic. Schema (a) is just the "flat structure" rule which expands a sentence as a lexical head together with all its subcategorized dependents (including the subject). This is the rule which sanctions sentences in so-called "non-configurational" languages (Warlpiri is the standard example, but I would also include Japanese here), root sentences in V-S-O languages (such as Welsh), and inverted sentences headed by auxiliaries in English.

Schema (b) is the rule that expands a phrase as a lexical head together with all its subcategorized dependents except for the last one (the subject). This is the rule responsible for VPs in languages like Chinese and English (I think the question as to whether Japanese has any VPs is still open).

Schema (c) is essentially "S $\rightarrow$ NP VP", though it is also responsible for clause-like structures headed by predicative APs or PPs ("small clauses" or "predicative adjunct clauses"). Together with Schema (b), it is responsible for "canonical" (non-inverted) sentences in English, for sentences in general in Chinese, and often for non-root (or non-finite) clauses in V-S-O languages.

10. Logical and Computational Considerations

In the remainder of this paper, I will sketch how the various components introduced above—lexical entries, principles of grammar, and schematic rules—are assembled into a formal theory. Thus far I have said quite a bit about linguistic objects (mostly signs) and the feature structures used to model them, and I have informally presented a number of principles and rules that are supposed to delimit the "admissible" feature structures, the ones that model actual signs. But to formalize all this, we need a logical language for expressing constraints on feature structures, the bottom corner of the triangle in Fig. 8. (This formal logic and its semantics are treated more fully in Pollard, 1988a, and Pollard & Moshier, in press.)

The logic I will use here is a slight variant of the one proposed by Rounds & Kapser (1986), augmented to include sort symbols, so I call it SRK (sorted Rounds-Kasper logic). The syntax of SRK is defined as in Fig. 26.

Here it is assumed that we are given in advance two finite disjoint sets, the set of sorts and the set of attribute labels. The set of sorts is also assumed to be equipped with a partial ordering (more precisely, a meet-semilattice structure, so that any nonempty set of sorts with

1. a is a formula ($a \in Sorts$),

2. *Bottom* and *Top* are formulas;

3. if ϕ and ψ are formulas then $\phi \wedge \psi$ is a formula;

4. if ϕ and ψ are formulas then $\phi \vee \psi$ is a formula;

5. if u and v are paths, $u \doteq v$ is a formula;

6. if $l \in Labels$ and ϕ is a formula then $l : \phi$ is a formula.

Figure 26: The Syntax of *SRK*

1. $A \models a$ iff $a \sqsubseteq \alpha(q_0)$;

2. $A \models Bottom$ always and $A \models Top$ never;

3. $A \models \phi \wedge \psi$ iff $A \models \phi$ and $A \models \psi$;

4. $A \models \phi \vee \psi$ iff $A \models \phi$ or $A \models \psi$;

5. $A \models u \doteq v$ iff $\delta(q_0, u) = \delta(q_0, v)$;

6. $A \models l : \phi$ iff $A/l \models \phi$.

Figure 27: The Semantics of *SRK*

an upper bound has a least upper bound, or join). The sorts, of course, are just names of the basic types of objects in the empirical domain (sign, word, phrase, category, parameter, etc.). The intuition behind the sort ordering is that one sort is greater than a second if the type it denotes is a subtype of the type denoted by the second; that is, "greater" means "more informative" or "more restrictive". For example, *sign* $\sqsubseteq$ *phrase*, *word* $\sqsubseteq$ *phrase*, *boolean* $\sqsubseteq$ +, and *list* $\sqsubseteq$ *nelist*.

The interpretation of formulas in SRK is given by defining a satisfaction relation between sorted feature structures and formulas. This definition is given in Fig. 27.

Here a sorted feature structure A is thought of as a 4-tuple $< Q, q_0, \delta, \alpha >$, where Q is the set of nodes, q_0 is the root node, δ is the (partial) transition function from node-path pairs to nodes, and α is the function that assigns sorts to nodes. (Equivalently, a sorted feature structure is a finite-state machine with sort symbols assigned to its states.) A path is just a finite sequence of attribute labels, and the notation A/l refers to the substructure of A reached by following the arc labelled by l.

A. Equations for Distributive Lattices with Top and Bottom

$\phi \vee Top = \phi$ (Top)
$\phi \wedge Top = Top$
$\phi \vee Bottom = Bottom$ (Bottom)
$\phi \wedge Bottom = \phi$
$\phi \vee \psi = \psi \vee \phi$ (Commutativity)
$\phi \wedge \psi = \psi \wedge \phi$
$(\phi \vee \psi) \vee \xi = \phi \vee (\psi \vee \xi)$ (Associativity)
$(\phi \wedge \psi) \wedge \xi = \phi \wedge (\psi \wedge \xi)$
$\phi \vee \phi = \phi$ (Idempotence)
$\phi \wedge \phi = \phi$
$(\phi \vee \psi) \wedge \phi = \phi$ (Absorption)
$(\phi \wedge \psi) \vee \phi = \phi$
$(\phi \vee \psi) \wedge \xi = (\phi \wedge \xi) \vee (\psi \wedge \xi)$ (Distributivity)
$(\phi \wedge \psi) \vee \xi = (\phi \vee \xi) \wedge (\psi \vee \xi)$

B. Equations for Path Equality

$Bottom = (\epsilon \doteq \epsilon)$ (Trivial Reflexivity)
$(uv \doteq w) = (uv \doteq w) \wedge (u \doteq u)$ (Restricted Reflexivity)
$uv : a = uv : a \wedge (u \doteq u)$
$(u \doteq v) = (v \doteq u)$ (Symmetry)
$(u \doteq v) \wedge (v \doteq w) = ((u \doteq v) \wedge (v \doteq w)) \wedge (u \doteq w)$ (Transitivity)
$(u \doteq v) \wedge u : a = ((u \doteq v) \wedge u : a) \wedge v : a$ (Restricted Substitutivity)
$(u \doteq v) \wedge (w \doteq vy) = ((u \doteq v) \wedge (w \doteq vy)) \wedge (w \doteq uy)$

C. Equations for Labels and Sorts

$l : Top = Top$ (Labels)
$l : \phi \wedge l : \psi = l : (\phi \wedge \psi)$
$l : \phi \vee l : \psi = l : (\phi \vee \psi)$
$l : (u \doteq v) = (lu \doteq lv)$
(equations for joins in the sort ordering $\sqsubseteq$)

Figure 28: Equational Calculus for *SRK*

a. $sign = word \vee phrase$

b. $word = \lambda_1 \vee \ldots \vee \lambda_l$

c. $phrase = (\phi_1 \vee \ldots \vee \phi_m) \wedge \psi_1 \wedge \ldots \wedge \psi_n$

Figure 29: Some definitions in a formal HPSG grammar

An SRK formula is called *normal* if it is a disjunction of disjunction-free formulas. It is easy to see that for any sorted feature structure A, there is an essentially unique disjunction-free formula that fully describes it, the *canonical formula of A*, denoted by $can(A)$. In general, of course, an SRK formula may be satisfied by (possibly infinitely) many feature structures. But as Rounds and Kasper have shown, for a given satisfiable formula ϕ, the set $Sat(\phi)$ of feature structures which satisfy ϕ is finitely generated, in the sense that there are finitely many feature structures $A_1, \ldots, A_n$, none subsuming another, such that $Sat(\phi)$ is precisely the set of feature structures subsumed by (at least) one of the A_i. In fact, we can algorithmically reduce ϕ to an equivalent normal formula whose disjuncts are just the $can(A_i)$; if ϕ is not satisfiable, then it can be reduced by the same algorithm to Top.

The actual algorithm depends on an equational calculus (Fig. 28) which is sound and complete for SRK under the given semantics; this is essentially the calculus given by Rounds and Kasper, with extra equations to encode the joins of the sorts. Formulas are reduced by using the equations as rewrite rules. In practice, of course, this is not usually a very efficient way to solve equations, but it shows that the problem in principle has a finite solution.

We can now formalize our linguistic theory as a grammar using SRK. The basic idea here is that a grammar for a particular natural language is a set of simultaneous recursive definitions for all the basic linguistic types. We might begin by defining a sign to be either a word or a phrase, as in Fig. 29a. The definition of *word*, in turn, is just a big normal formula (29b), where each disjunct λ_i is the canonical formula of some lexical feature structure (such as the one in Fig. 5); that is, the lexicon essentially has to be listed. What about the definition of phrase? Well, a feature structure will correspond to a well-formed phrase just in case it satisfies *all* of the (universal and language-specific) well-formedness principles and *one* of the phrase-structure schemata of the language in question. That is, if each principle is encoded by one of the formulas $\psi_1, \ldots, \psi_n$, and each phrase-structure schema is encoded by one of the formulas $\phi_1, \ldots, \phi_m$, then the definition of *phrase* should simply be the formula (29c). Of course the grammar must contain a definition for every basic type. The way recursion comes into the picture is that, in general, the sort symbols can occur on the right-hand sides of the definitions; for example, the symbol *sign* occurs in the definition of *sign*.

Here, of course, some of the ψ_j in (29c) will be encodings of universal principles such as the HFP, the Subcategorization Principle, and the binding theory. The first two of these, given in (30), should be compared with the informal statements in (14).

(30) SRK Encodings of Head Feature Principle and Subcategorization Principle

a. SYNTAX:HEAD $\doteq$ DTRS:HEAD-DTR:SYNTAX:HEAD

b. *append*(DTRS:HEAD-DTR:SYNTAX:SUBCAT, DTRS:COMP-DTRS, SYNTAX:SUBCAT)

In the case of English, one of the ψ_j in (28c) will encode the English version of the "S → NP VP" phrase structure schema (stated informally in (25c)); this is shown in (31). Notice that the English version includes not only the universal information about immediate dominance, but also the fact that the phonological realization of the complement (i.e. the subject) precedes that of the head (the VP).

(31) SRK Encoding of English "S → NP VP" Phrase Structure Schema

SYNTAX:SUBCAT:*elist* ∧ DTRS:HEAD-DTR:*phrase* ∧

DTRS:COMP-DTRS:*nelist* ∧ DTRS:COMP-DTRS:REST:*elist* ∧

append(PHONOLOGY, DTRS:HEAD-DTR:PHONOLOGY, DTRS:COMP-DTRS:FIRST:PHONOLOGY)

By the way, in order for list manipulations such as those employed in (30) and (31) to make sense, some basic theory of lists also has to be included in the grammar. This is shown in (32). Notice that the recursive definition of *append* is actually a definition schema, with three path-valued parameters u, v, and w.

(32) Encoding of list theory in an HPSG grammar

a. $list = elist \vee nelist$

b. $nelist$ = FIRST:*bottom* ∧ REST:*list*

c. $append(u, v, w) = ((v : elist) \wedge (u \doteq w)) \vee$
$((u.\text{FIRST} \doteq v.\text{FIRST}) \wedge append(u.\text{REST}, v.\text{REST}, w))$

In Pollard, 1988a, I show that formal grammars of the kind sketched above actually have a well-defined denotational semantics,[9] and the denotation of the grammar can be computed by unifying each sort symbol on the right-hand side of the grammar with the definitions of all the sorts that subsume it, then successively repeating the process using the new "improved" definitions. In this context, "unification" corresponds to conjunction in SRK. This approach represents a synthesis of work on feature structure logic by Rounds & Kasper, 1986; the lattice-theoretic approach to knowledge representation of Ait-Kaci, 1984; and ideas on the denotational semantics of unification-based grammars due to Pereira & Shieber, 1984, and Moshier, 1988.

[9]The denotation of such a grammar is the least fixed point of a continuous transformation on a cartesian power of the Smyth powerdomain over the semilattice of all sorted feature structures.

In principle, this same process can be adapted as an all-paths HPSG parser. The trick here is to encode a parsing problem as the definition of a new sort called *input*. For example, (33) shows the encoding for the problem of parsing the string "Bill sneezed" as a finite sentence.

(33) Encoding of parsing problem for "Bill sneezed"

$$input = (\text{PHONOLOGY}{:}(\text{FIRST}{:}\text{bIl} \wedge \text{REST}{:}(\text{FIRST}{:}\text{snizd} \wedge \text{REST}{:}elist))) \wedge$$
$$(\text{SYNTAX}{:}((\text{SUBCAT}{:}elist) \wedge (\text{HEAD}{:}(\text{MAJ}{:}v \wedge \text{VFORM}{:}fin))))$$

If the grammar is a reasonable one, then successive expansion of such a definition with respect to the total grammar as sketched above terminates with a normal formula whose disjuncts are the SRK encodings of the possible parses.

Similarly, the effect of a generator can be obtained by specifying the SYNTAX and SEMANTICS:CONTENT of the *input* sort. In this case, however, we do not expect the process to terminate (since any message has infinitely many paraphrases); instead, the results will be enumerated in a stream.

11. Conclusion

As I noted above, a linguistic theory has to characterize what every language user knows, a system of linguistic types. But this is a rather crude criterion of adequacy. It is not enough only to say what the well-formed linguistic objects are, for that fails to account in any way for the possibility of language use. To reach that more ambitious goal, we need to explain how a language user can recognize in a finite amount of time that an actual utterance token, say "Look out for that truck!", is an instance of a certain type and not of some other type. That is, the theoretical model must also model the fact that a language user is an information-processing organism with the ability to *compute* the type of a linguistic object. This is an aspect of linguistic theory that has no analog in physics: a theory of celestial mechanics is a set of differential equations, but the theory is not required to reflect the computability of the solutions (indeed, the solutions need not be computable).

In HPSG, this key criterion, the computability of the linguistic types, is reflected by the fact that the set of parses of an input sentence is a finite (compact) object in a certain computational domain; that is to say, in principle it can be computed in finite time. It is in this sense that HPSG is a computational linguistic theory.

On the other hand, HPSG itself is inherently declarative rather than procedural; it does not specify a parsing algorithm. The successive unification procedure mentioned above is available in principle, but it is just one of many possible procedures that might be used; it is not part of the theory. The question remains whether a parsing algorithm can be developed for grammars of the general character described here that is efficient enough for the needs of practical computer-based natural language processing. I hope to address this question in future research.

References

Ait-Kaci, H. (1984) *A Lattice-Theoretic Approach to Computation Based on a Calculus of Partially Ordered Types.* Ph.D. dissertation, Department of Computer Science, Moore School, University of Pennsylvania.

Barwise, J. and J. Perry (1983) *Situations and Attitudes.* Cambridge, MA: Bradford Books, MIT Press.

Bresnan, J. (ed.) (1982) *The Mental Representation of Grammatical Relations.* Cambridge, MA: MIT Press.

Chomsky, N. (1957) *Syntactic Structures.* The Hague, NL: Mouton.

Chomsky, N. (1981) *Lectures on Government and Binding.* Dordrecht, NL: Foris.

Chomsky, N. (1986) *Knowledge of Language: Its Nature, Origin and Use.* New York, NY: Praeger.

Chomsky, N. (1987) *Barriers.* Cambridge, MA: MIT Press.

Clements, G.N. (1985) The Geometry of Phonological Features. In *Phonology Yearbook* 2, 225–252.

Fenstad, J.-E., P.-K. Halvorsen, T. Langholm and J. van Venthem (1987) *Situations, Language, and Logic.* Studies in Linguistics and Philosophy Series, Dordrecht, NL: Reidel.

Gazdar, G., K. Klein, G.K. Pullum and I.A. Sag (1985) *Generalized Phrase Structure Grammar.* Cambridge, UK: Blackwell and Cambridge, MA: Harvard University Press.

Gazdar, G., G.K. Pullum, R. Carpenter, E. Klein, T. Hukari and R. Levine (1988) Category Structures. *Computational Linguistics*, 14.1, 1-19. Also published as Report No. CSLI-87-102, Stanford, CA: CSLI.

Johnson, M. (1987) *A Logic of Attribute-Value Structures and the Theory of Grammar.* Ph.D. dissertation, Stanford University.

Karttunen, L. (1986a) D-PATR: A Development Environment for Unification-Based Grammars. Report No. CSLI-86-68, Stanford, CA: CSLI.

Kasper, R. and W. Rounds (1986) A Logical Semantics for Feature Structures. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics.*

Kay, M. (1979) Functional Grammar. In C. Chiarello et al. (eds.), *Proceedings of the Fifth Annual Meeting of the Berkeley Linguistics Society.*

Kay, M. (1985) Parsing in Functional Unification Grammar. In D. Dowty, L. Karttunen and A. Zwicky (eds.), *Natural Language Parsing*, Cambridge, UK: Cambridge University Press.

Keenan, E. and B. Comrie (1977) Noun Phrase Accessibility and Universal Grammar. *Linguistic Inquiry, 8*: 63-99.

Moshier, D. (1988) *Extensions to Unification Grammar for the Description of Programming Languages.* Ph.D. dissertation, University of Michigan.

Moshier, D. and W. Rounds (1987) A Logic for Partially Specified Data Structures. In *Proceedings of the 14th Annual Conference on Principles of Programming Languages.*

Pereira, F. (1987) Grammars and Logics of Partial Information. Technical Note 420, Menlo Park, CA: SRI International.

Pereira, F. and S. Shieber (1984) The Semantics of Grammar Formalisms Seen as Computer Languages. In *Proceedings of COLING 84*, 23–129.

Pollard, C. (1988a) Sorts in Unification-Based Grammar and What They Mean. To appear in M. Pinkal and B. Gregor (eds.), *Unification in Natural Language Analysis.*

Pollard, C. (1988b) NP Typology and the Binding Theory. Manuscript, Carnegie Mellon University. To appear as a CSLI technical report.

Pollard, C. (1989) The Typology of Chinese Noun Phrases. Paper presented in colloquium at Tsing Hua University, Taipei, Taiwan, October 1988.

Pollard, C. (in preparation) An Integrated Syntactic-Semantic Approach to Unbounded Dependencies. To appear in *Linguistics & Philosophy.*

Pollard, C. and M.D. Moshier (in press) Unifying Partial Descriptions of Sets. In P. Hanson (ed.), *Information, Language, and Cognition, Vancouver Studies in Cognitive Science I.*

Pollard, C. and I.A. Sag (1987) *Information-Based Syntax and Semantics. Volume 1: Fundamentals.* Lecture Notes Series No. 13, Stanford, CA: CSLI.

Pollard, C. and I.A. Sag (1988) An Information-Based Theory of Agreement. In D. Brentari, G. Larson and L. Macleod (eds.), *Proceedings of the 24th Regional Meeting of the Chicago Linguistic Society*, vol. 2: Parasession on Agreement in Grammatical Theory, Chicago, IL: University of Chicago.

Rounds, W. and R. Kasper (1986) A Complete Logical Calculus for Record Structures Representing Linguistic Information. In *Proceedings of the IEEE Symposium on Logic in Computer Science*, June 1986.

Sag, I.A., R. Kaplan, L. Karttunen, M. Kay, C. Pollard, S. Shieber and A. Zaenen (1986) Unification and Grammatical Theory. In *Proceedings of the Fifth West Coast Conference on Formal Linguistics*, Stanford, CA: Department of Linguistics.

Sag, I.A. and C. Pollard (ms.) A Semantic Theory of Obligatory Control. Manuscript, Stanford University and Carnegie Mellon University.

Shieber, S. (1984) The Design of a Computer Language for Linguistic Information. In S. Shieber, L. Karttunen and F. Pereira (eds.), *Notes from the Unification Underground: A Compilation of Papers on Unification-Based Grammar Formalisms*, SRI Technical Note 327, Menlo Park, CA: SRI International.

Shieber, S. (1986) *An Introduction to Unification-Based Approaches to Grammar*, CSLI Lecture Notes 4, Stanford, CA: CSLI.

Shieber, S., H. Uszkoreit, J. Robinson and M. Tyson (1983) The Formalism and Implementation of PATR-II. In *Research on Interactive Acquisition and Use of Knowledge*, Menlo Park, CA: Artificial Intelligence Center, SRI International.

Smyth, M. (1978) Power Domains. *Journal of Computer Systems Science*, *16*, 23–36.

Thomason, R. (1987) Remarks on Linguistic Semantics. In A. Manaster-Ramer (ed.), *Mathematics of Language*, Amsterdam, NL and Philadelphia, PA: Benjamins.