1

Strong Generative Capacity in HPSG

CARL J. POLLARD[1]

1 Introduction

Grammars exist in the real world, more specifically the part of the world inside language knower's minds; utterance tokens are events, also in the real world but external (at least in part) to minds. Formal linguists create mathematical idealizations of both these things. A formal grammar, a certain kind of logical theory, is a mathematical idealization of the mental grammar; and utterance tokens which are judged grammatical by native language knowers are idealized as certain mathematical structures, in such a way that distinct utterances which are linguistically indistinguishable, e.g. two utterances of *Poor John ran away* are idealized as isomorphic structures. What I think ties all these things together is that, given an appropriate scheme of semantic interpretation for formal grammars, and assuming it is a good grammar, the mathematical structures which idealize the grammatical utterance tokens will be (certain kinds of) *models* of the grammar. We can state this equivalently in terms of prediction: a formal grammar is a scientific theory which predicts that an utterance token will be judged grammatical if and only if it is (or more precisely, is idealized as) a model of the grammar (more specifically, a model which is *singly generated*, in a sense to be explained below).

On the other hand, generative grammarians are accustomed to think-

[1] I owe a profound debt of gratitude to Paul King, whose ongoing critique of the HPSG research program has played a central role in the development of the ideas expressed in this paper. King, however, does not necessarily endorse the approach to the notion of generation that I take in this paper.

ing of a formal grammar as *generating* a set of structures, called the *strong generative capacity* (SGC) of the grammar? This set is usually considered to have the following two properties: (1) no two of its members are structurally isomorphic; and (2) if the grammar is making good predictions, then utterance tokens whose structures are isomorphic to members of the SGC will be judged grammatical.

Now consider the following two questions. First, recall that immediately above we considered two token utterances of *Poor John ran away.* The question is: precisely what, if anything, is this mysterious entity *Poor John ran away* of which our two token utterances are instances, and should we (as linguists) care? The second question is: what, if anything, is the SGC of a grammar. In this paper, we propose answers to these questions. Most of the paper is formal in character, but the intuition behind the formalities can be stated simply along the following lines.

As for the first question, one may be tempted to say that *Poor John ran away* is a type, or property, of which distinct token utterances are instances. Alas, in what sense such types exist, if at all, is a question that is answered differently in different ontologies. Fortunately, though, nothing about the nature of linguistic theory requires us to take a stand on this question one way or the other. On the other hand, it would be nice to have a name for the isomorphism class of structures which includes (the structures that idealize) all these tokens; we could perfectly well call it the *Poor John ran away* isomorphism class without accepting the reality of a *Poor John ran away* type. For that matter, there is a well-known technique—to be explained below—for selecting from an isomorphism class of structures (at least, for the kinds of structures we are dealing with), a canonical representative. I am going to take the liberty of using *Poor John ran away* as the name of the canonical representative of the isomorphism class discussed immediately above. Thus, according to the practice I am adopting, *Poor John ran away* is a canonical representative of a certain isomorphism class of singly generated models of our formal grammar.

Second, what does our grammar generate? Well, *Poor John ran away,* among other things. More generally, we will *define* the **strong generative capacity** (SGC) of a grammar to be the set of canonical representatives of the isomorphism classes of singly generated models of the grammar. This notion of SGC satisfies the two criteria mentioned above, without entailing any ontological commitment to utterance types.

2 Then and Now

In early HPSG (Pollard and Sag 1987, hereafter, HPSG'87), the formalization of grammar was based on the use of 'intuitionistic' feature structures to model *partial information* about linguistic entities. On that approach, pioneered by Pereira and Shieber 1984, Kasper and Rounds 1986, and Moshier 1988, feature structures are essentially canonical forms for logical equivalence classes of linguistic descriptions. The linguistic descriptions themselves (which may be informal attribute-value matrices or expressions in a formal feature logic) can be thought of as preordered by relative informativeness (with more informative elements lower in the preorder); here the descriptions are analogous to formulas in an intuitionistic propositional logic. Identifying informationally equivalent descriptions (which is the same as forming the Lindenbaum algebra of the feature logic), we obtain a Heyting algebra H, and it is elements of this algebra that were employed in early HPSG as lexical entries, grammar rules, and well-formedness principles. Now suppose that, within this algebra, we limit our attention to the subset of *coprime* elements, i.e. the equivalence classes of noncontradictory nondisjunctive formulas, but now ordered with the (restriction of the) dual ordering (with relatively more informative elements higher). Then the resulting partial order turns out to be isomorphic to the set of finite feature structures under the subsumption ordering. So in a precise mathematical sense, feature structures and feature descriptions are interdefinable. [2]

By contrast, and in accordance with the general trend in feature logic from the late 1980's on (cf. Johnson 1988, Smolka 1988, King 1989), the HPSG of Pollard and Sag 1994 (hereafter, HPSG'94), distinguishes sharply between feature descriptions and feature structures. On this more recent approach, feature structures are now employed only as (total) mathematical idealizations of linguistic entities, and the information ordering holds only among descriptions. As explained below, descriptions are employed in grammars in order to pick out that subset of the feature structures that will be regarded as (idealizations of) possible

[2] To be precise, the relationship is as follows: the set of (abstract) feature structures (including ones with infinite node sets) under the subsumption ordering is a (Scott) domain whose set of Scott compact-open subsets (i.e. finite unions of principal filters generated by finite feature structures) ordered by inclusion is isomorphic to H. In the other direction, the feature structure domain is recoverable is as the *spectrum* (i.e. the set of prime ideals) of H. Thus we have a Stone duality (in the sense of Johnstone 1982:66) between the feature structure domain and the (Lindenbaum algebra of) the feature logic. In this setting, a grammar is a finite set of sort-description pairs, and its denotation is determined by a certain fixed-point construction on a cartesian power of the feature structure domain. For details, see Pollard and Moshier 1990 and Pollard to appear.

well-formed linguistic entities. Crucially, in order for negative and implicative constraints to work as intended, the underlying feature logic must now be classical (in contrast with the intuitionistic logic of the earlier approach). Now very roughly speaking, the relationship between an HPSG grammar and its strong generative capacity is analogous to the relationship between a first-order theory and its models. There are, however, some subtle differences, and it is something of an embarrassment that there is as yet no place in the literature where the relationship between an HPSG'94 grammar and its SGC is given a characterization. In the course of this paper we intend to fill that gap.

The remainder of this paper is organized as follows. In section 3, we introduce a particular formal logic, SRL, which will be used for writing linguistic descriptions and grammars, together with a certain kind of algebraic interpretation of descriptions and grammars which is analogous in certain respects to the standard set-theoretic interpretation of classical logic. However, the models yielded by this kind of interpretation do not correspond directly to the usual linguistic notion of SGC. Section 4 digresses briefly from our main agenda to explain the relationship between the formal grammars of section 2 and informal grammars along the lines of HPSG'94. In section 5, we explain in informal terms how to use the models of section 3 to construct a more satisfactory mathematical embodiment of the notion of SGC. Finally, in section 6, we synopsize the mathematics that legitimates the informal approach of section 5.

3 Descriptions, Theories, Interpretations, and Models

In feature-based grammatical frameworks (such as HPSG), linguistic entities are idealized of as structured objects, i.e. things with identifiable parts. An obvious example is that the constituents of a phrase can be regarded as parts. A second example is that the different 'levels of representation' or aspects of linguistic structure (phonology, morphology, syntax, semantics, pragmatics, etc.) of a linguistic expression can be viewed as parts of the expression; such a view is particularly appropriate within nonderivational frameworks (such as HPSG) where the different levels are considered to 'exist in parallel' (as opposed to one or more of the levels being sequentially derived by operating on or transforming an earlier level). A third example is that the morphosyntactic features of a word (such as the case of a noun or the inflectional form of a verb) can be considered a part of the word. In HPSG all these notion of part (inter alia) play a significant role.

The two fundamental ontological assumptions that give HPSG theory much of its distinct flavor are (1) the closed world assumption, and (2) the assumption of total well-typedness. The closed world assumption is simply that there is a fixed set of disjoint ontological categories (species) which partition the universe of linguistic entities. Total well-typedness means that (a) for each species, there is a fixed inventory of identifiable immediate subparts (features) such that every entity of that species has those and only those features; and (b) for each species and for each feature appropriate to that species, there is a fixed set of species to which the corresponding immediate subpart can belong. Simply put: different kinds of things (species) have different kinds of identifiable parts (features), which in turn must be of certain kinds (other species). Thus, e.g. linguistic expressions have various levels of representation; phrases (as opposed to words) have syntactic immediate constituents, each of which must be a word or a phrase; and the parts of speech of different classes of expressions have different morphosyntactic features, etc. In order to talk about species and their features with precision, both in describing linguistic entities and in writing grammars that predict what linguistic entities are well-formed, we will employ a particular logic, SRL (King 1989), that is specially designed for this purpose. [3]

Syntactically, an SRL language is determined by specifying two disjoint sets of nonlogical symbols (species and features) and a notion of appropriateness that relates them to each other; semantically, for a given SRL language, there will be a class of interpretations each of which provides meanings for the nonlogical symbols and for descriptions (formulas of the language). Unlike the case of many familiar logics (e.g. the first-order predicate calculus), an SRL description, when interpreted, is taken to denote not a truth value but rather a subset of the points in the interepretation. Let us make this more precise.

A **signature** is a triple $\Sigma = \langle \mathsf{S}, \mathsf{F}, \mathsf{A} \rangle$, where S and F are sets, whose members are called **species** and **features** respectively, and A is a total function from $\mathsf{S} \times \mathsf{F}$ to $Pow(\mathsf{S})$, called the **appropriateness** function. An **interpretation** of Σ is a triple $I = \langle U, S, F \rangle$, where U is a set, S is a total function from U to S, F is a total function from F to the set of partial functions from U to U, and for each $\varphi \in \mathsf{F}$, for each $\upsilon \in U$,

$F(\varphi)(\upsilon)$ is defined iff $\mathsf{A}(S(\upsilon), \varphi) \neq \emptyset$, and

[3] SRL is very similar in character to the attribute-value logic (AVL) of Johnson 1988, and to the feature logic (FL) of Smolka 1988; for detailed comparison, see King 1989. The most significant difference between SRL and these other options is that the fundamental ontological assumptions (closed world and total well-typedness) are directly built into the notion of interpretation (see below), i.e. there are no interpetations for SRL in which closed world and total well-typedness do not hold. In addition, SRL is essentially equivalent to the typed feature logic of Carpenter 1992 as long as the latter is equipped with axioms of inequation-resolvedness, total well-typedness, and closed world, and as long as feature structures with infinite node sets are permitted.

if $F(\varphi)(\upsilon)$ is defined then $S(F(\varphi)(\upsilon)) \in \mathsf{A}(S(\upsilon), \varphi)$.

The three components of the interpretation are called its **universe**, **species interpretation function**, and **feature interpretation function**, respectively; and the members of the universe are called its **objects**. [4] Informally, each species denotes a set of objects, each object is in the denotation of exactly one species, and S assigns to each object the unique species to whose denotation the object belongs; thus species σ denotes the set $\{\upsilon \in U \mid S(\upsilon) = \sigma\}$. Each feature denotes a partial function from objects to objects, and F assigns to each feature the partial function it denotes; thus the feature φ denotes the partial function $F(\varphi)$.

The appropriateness function encodes—and the definition of an interpretation enforces—a strict relationship between the denotations of species and features: if $\mathsf{A}(\sigma, \varphi) = \emptyset$ then the domain of $F(\varphi)$ does not intersect the denotation of σ; but if $\mathsf{A}(\sigma, \varphi) \neq \emptyset$ then that domain includes the denotation of σ and maps it into the union of the denotations of the species in $\mathsf{A}(\sigma, \varphi)$. For example, one would put $\mathsf{A}(\textit{monarchy}, \textsc{monarch}) = \{\textit{king}, \textit{queen}\}$ to indicate that each monarchy has a monarch that is either a king or a queen, and put $\mathsf{A}(\textit{republic}, \textsc{monarch}) = \emptyset$ to indicate that each republic has no monarch. For convenience, we henceforth assume that none of the logical symbols $:$, $\sim$, $\approx$, $\neg$, $\wedge$, $\vee$, $\rightarrow$, [and] is a species or a feature. In the remainder of this paper, we will assume a fixed signature $\Sigma = \langle \mathsf{S}, \mathsf{F}, \mathsf{A} \rangle$.

We now define by simultaneous recursion the sets of **terms**, T_Σ, and **descriptions**, D_Σ. [5] These are the smallest sets such that

$: \in \mathsf{T}_\Sigma$,

for each $\tau \in \mathsf{T}_\Sigma$, for each $\varphi \in \mathsf{F}$, $\tau\varphi \in \mathsf{T}_\Sigma$,

for each $\tau \in \mathsf{T}_\Sigma$, for each $\sigma \in \mathsf{S}$, $\tau \sim \sigma \in \mathsf{D}_\Sigma$,

for each $\tau_1 \in \mathsf{T}_\Sigma$, for each $\tau_2 \in \mathsf{T}_\Sigma$, $\tau_1 \approx \tau_2 \in \mathsf{D}_\Sigma$,

for each $\delta \in \mathsf{D}_\Sigma$, $\neg\delta \in \mathsf{D}_\Sigma$,

for each $\delta_1 \in \mathsf{D}_\Sigma$, for each $\delta_2 \in \mathsf{D}_\Sigma$, $[\delta_1 \wedge \delta_2] \in \mathsf{D}_\Sigma$,

for each $\delta_1 \in \mathsf{D}_\Sigma$, for each $\delta_2 \in \mathsf{D}_\Sigma$, $[\delta_1 \vee \delta_2] \in \mathsf{D}_\Sigma$, and

for each $\delta_1 \in \mathsf{D}_\Sigma$, for each $\delta_2 \in \mathsf{D}_\Sigma$, $[\delta_1 \rightarrow \delta_2] \in \mathsf{D}_\Sigma$.

For a fixed interpretation $I = \langle U, S, F \rangle$ of Σ, we now explain how the

[4]Mathematically, an interpretation is a sorted unary partial algebra, with species as sorts and feature denotations as operations, with special conditions on the domains and ranges of the operations. For this reason they are also often known as **feature algebras**.

[5]Terms and descriptions are also often called **paths** and **formulas**, respectively.

terms and descriptions are interpreted. First, the **term interpretation** function, T_I, maps terms to partial functions from U to U as follows:

for each $\upsilon \in U$,

$T_I(:)(\upsilon)$ is defined and $T_I(:)(\upsilon) = \upsilon$,

for each $\tau \in \mathsf{T}_\Sigma$, for each $\varphi \in \mathsf{F}$, for each $\upsilon \in U$,

$T_I(\tau\varphi)(\upsilon)$ is defined iff $T_I(\tau)(\upsilon)$ is defined and $F(\varphi)(T_I(\tau)(\upsilon))$ is defined, and

if $T_I(\tau\varphi)(\upsilon)$ is defined then $T_I(\tau\varphi)(\upsilon) = F(\varphi)(T_I(\tau)(\upsilon))$

Informally, : denotes the identity function on the universe, and other terms, which are nonnull strings of features, denote the function composition of the functions denoted by those features.

Next, the **description interpretation function**, D_I, maps descriptions to subsets of the universe as follows:

for each $\tau \in \mathsf{T}_\Sigma$, for each $\sigma \in \mathsf{S}$,

$$D_I(\tau \sim \sigma) = \left\{ \upsilon \in U \;\middle|\; \begin{array}{l} T_I(\tau)(\upsilon) \text{ is defined and} \\ S(T_I(\tau)(\upsilon)) = \sigma \end{array} \right\},$$

for each $\tau_1 \in \mathsf{T}_\Sigma$, for each $\tau_2 \in \mathsf{T}_\Sigma$,

$$D_I(\tau_1 \approx \tau_2) = \left\{ \upsilon \in U \;\middle|\; \begin{array}{l} T_I(\tau_1)(\upsilon) \text{ is defined,} \\ T_I(\tau_2)(\upsilon) \text{ is defined, and} \\ T_I(\tau_1)(\upsilon) = T_I(\tau_2)(\upsilon) \end{array} \right\},$$

for each $\delta \in \mathsf{D}_\Sigma$, $D_I(\neg\delta) = (U \setminus D_I(\delta))$,

for each $\delta_1 \in \mathsf{D}_\Sigma$, for each $\delta_2 \in \mathsf{D}_\Sigma$, $D_I([\delta_1 \wedge \delta_2]) = D_I(\delta_1) \cap D_I(\delta_2)$,

for each $\delta_1 \in \mathsf{D}_\Sigma$, for each $\delta_2 \in \mathsf{D}_\Sigma$, $D_I([\delta_1 \vee \delta_2]) = D_I(\delta_1) \cup D_I(\delta_2)$,

for each $\delta_1 \in \mathsf{D}_\Sigma$, for each $\delta_2 \in \mathsf{D}_\Sigma$,

$D_I([\delta_1 \rightarrow \delta_2]) = (U \setminus D_I(\delta_1)) \cup D_I(\delta_2)$.

Informally, $\tau \sim \sigma$ denotes those objects from which an object of species σ is reached by following the path τ; $\tau_1 \approx \tau_2$ denotes those objects from which the paths τ_1 and τ_2 lead to the same object; and the boolean connectives work as in classical propositional logic.

It is crucial to bear in mind that descriptions denote subsets of the interpretation, not truth values. To put it another way, unlike the situation with (say) first-order logic, we speak not of truth in an interpretation, but rather of truth at an object in an interpretation. That is, for a fixed interpretation, we say that a description is **true** at an object of the interpretation just in case the object is in the denotation of the description.

We extend the notions of denotation and truth to **theories** (sets of descriptions) in the obvious way: the denotation of a theory θ, $\Theta_I(\theta)$, is the intersection of the denotations of the descriptions in θ, i.e.

$$\text{for each } \theta \subseteq \mathsf{D}_{\Sigma},\ \Theta_I(\theta) = \left\{ \upsilon \in U \,\middle|\, \text{for each } \delta \in \theta,\ \upsilon \in D_I(\delta) \right\},$$

and a theory is **true** at an object just in case each of its descriptions is. (An important special case of this is the **theory of** an object, which is defined as the set of all descriptions true at that object.) And an interpretation is called a **model** of a theory just in case the denotation of the theory (relative to that interpretation) is the whole universe. Equivalently, a model of a theory is an interpretation such that the theory is true at every object. Finally, a **grammar** is a pair $\langle \Sigma, \theta \rangle$, where Σ is a signature and $\theta \subseteq \mathsf{D}_{\Sigma}$. [6]

It might be expected that a model of the grammar (or, more precisely, of the grammar's theory) could be employed as the SGC of the grammar Such a notion of SGC has the virtue of mathematical simplicity, but it suffers from a number of defects. First, there can be many models of a given grammar, and these need not all be isomorphic. Second, some models of a grammar are 'too small', in the sense that the natural language whose grammar we are trying to write might have some grammatical expressions that do not correspond to any object of the model. And third, models may differ from each other in ways that make no linguistic difference, e.g. the number of objects in the model which correspond to the sentence *Poor John ran away.* There is a clear intuitive sense in which this should not matter linguistically; we want a notion of SGC which abstracts away from how many 'instances' or 'tokens' of a given expression there are, and instead is sensitive only isomorphism classes. In section 5 we will address these problems, and present an informal argument that, as far as SGC is concerned, it is not really the models of the grammar that are of interest, but rather a certain set of feature structures associated with certain 'sufficiently large' models (viz. the **exhaustive** models.) But first, we digress briefly to explain the connection between the formal notion of grammar just presented and the informal notion of grammar in the sense in which it is usually employed in HPSG practice.

4 Formal and Informal Grammars

According to Pollard and Sag 1994, a HPSG grammar

"consists of a *sort hierarchy* and a set of *principles.* The sort hier-

[6] Because we are concerned primarily with issues of grammar interpretation in this paper, we have ignored purely logical and computational aspects of SRL. SRL has a sound and complete logic (see King 1989), and it is decidable whether an arbitrary term is satisfiable, i.e. whether there is some object in some model at which the term is true (see Kepser 1994). For further work on the logical and computational aspects of SRL see Gerdemann and King 1994, Aldag 1997, King and Simov to appear, and King et al. to appear.

archy is presented as a taxonomic tree, with the root labelled *object* (the sort of all linguistic entities with which the grammar deals). For each local tree in the hierarchy, the sorts $\sigma_1, \ldots, \sigma_n$, which label the daughters, partition the sort σ, which labels the mother; that is, they are necessarily disjoint subsorts of σ that exhaust σ. For two sorts σ and τ, τ is a *subsort* of σ if and only if it is dominated by σ; sorts that label terminal nodes are called *maximal* (in the sense of maximally informative or maximally specific). A *feature declaration* of the form

$$\sigma\colon \begin{bmatrix} \mathrm{F_1}\ \tau_1 \\ \ldots \\ \mathrm{F_n}\ \tau_\mathrm{n} \end{bmatrix}$$

where $\sigma, \tau_1, \ldots, \tau_n$ are sorts and $\mathrm{F}_1, \ldots, \mathrm{F_n}$ are feature labels, signifies that for each $i = 1, \ldots, n$, (1) the feature $\mathrm{F_i}$ is appropriate for all objects of sort σ, and (2) for any such object, the value of the $\mathrm{F_i}$ feature must be an object of sort τ_i.

If sorts σ_1 and σ_2 bear the declarations $[\mathrm{F}\,\tau_1]$ and $[\mathrm{F}\,\tau_2]$ for the same feature F and σ_2 is a subsort of σ_1, then τ_2 must be a subsort of τ_1. A sort inherits the feature declarations of its supersorts; hence any feature that is defined for a given sort is defined for all of that sort's subsorts. By convention, the features that are declared for a maximal sort (including those inherited from its supersorts) are the only features defined for that sort. A special case is that of *atomic* sorts or simply *atoms*, maximal sorts for which no features are defined." (pages 395 and 396)

Moreover,

"The (finite) set of all sort symbols is assumed to be partially ordered, with sort symbols corresponding to more inclusive types lower in the ordering. For example, the sort *sign* is ordered below the sort *phrase* or *word* because signs include both phrases and words; we say, for example, that *word* is a *subsort* of *sign* and *accusative* is a subsort of *case*." (page 17)

Notice two points. First, the sort hierarchy is a finite partial order, sort *object* is a supersort of all other sorts, *object* denotes all linguistic objects, and the denotations of the immediate subsorts of a nonmaximal sort partition the denotation of the sort; therefore the maximal sorts are like species in that each linguistic object is in the denotation of one and only one maximal sort. Secondly, whereas for each nonmaximal sort and feature, the denotation of the feature can be defined on all objects in the denotation of the sort (as with feature PHONOLOGY and sort *sign*), on some but not all objects in the denotation of the sort (as with feature DAUGHTERS and sort *sign*), or on no objects in the denotation of the sort (as with feature PERSON and sort *sign*), the maximal sorts are like species in that for each maximal sort and feature, the denotation of the feature can be defined only on all objects in the denotation of the sort (as with feature DAUGHTERS and sort *phrase*), or on no objects in the denotation of the sort (as with feature PERSON and sort *phrase*). In fact, each maximal HPSG sort is an SRL species, and each nonmaximal HPSG sort is the disjunction of its maximal subsorts. For example, maximal HPSG sorts *phrase* and *word* are SRL species, and nonmaximal HPSG sort *sign* is the disjunction of *phrase* and *word*. Each HPSG feature is an SRL feature, and an SRL appropriateness function A can express the HPSG feature declarations for the maximal sorts: for each maximal sort σ, for each feature φ, if σ bears feature declaration $[\varphi\,\varsigma]$ then $\mathsf{A}(\sigma,\varphi)$ is the set of maximal subsorts of ς else $\mathsf{A}(\sigma,\varphi) = \emptyset$. The HPSG feature declarations for the nonmaximal sorts are inferable from A.[7] Thus, the signature component of an SRL grammar can express the sort hierarchy component of a HPSG grammar. Moreover, the theory component of an SRL grammar can express the set of principles that constitute the other component of a HPSG grammar. For example, since the SRL expression of nonmaximal HPSG sort *headed-structure* is the disjunction of the SRL species *head-complement-structure*, *head-marker-structure*, *head-adjunct-structure* and *head-filler-structure*, the SRL expression of the Pollard and Sag 1994 head feature principle "In a headed phrase, the values of SYNSEM | LOCAL | CATEGORY | HEAD and DAUGHTERS | HEAD-DAUGHTER | SYNSEM | LOCAL | CATEGORY | HEAD are token-identical" (page 399) is the SRL description

$$\begin{bmatrix} \begin{bmatrix} :\sim \mathit{phrase} \;\wedge & \begin{bmatrix} :\text{DAUGHTERS} \sim \mathit{head\text{-}complement\text{-}structure} \\ \vee :\text{DAUGHTERS} \sim \mathit{head\text{-}marker\text{-}structure} \\ \vee :\text{DAUGHTERS} \sim \mathit{head\text{-}adjunct\text{-}structure} \\ \vee :\text{DAUGHTERS} \sim \mathit{head\text{-}filler\text{-}structure} \end{bmatrix} \end{bmatrix} \\ \rightarrow \;\; :\text{SYNSEM LOCAL CATEGORY HEAD} \approx \\ \quad :\text{DAUGHTERS HEAD-DAUGHTER SYNSEM LOCAL CATEGORY HEAD} \end{bmatrix}.$$

Though perhaps not obvious, it is true that SRL as it stands can encode recursively defined relations (e.g. append), by using the reification technique pioneered by Aït-Kaci 1984 (also known by its detractors as the 'junk slot' method); however this necessitates the introduction of features which do not correspond to the linguist's conception

[7] As pointed out to us by an anonymous referee, Pollard and Sag 1994 failed to make this point explicit. This rules out, for example, the possibility of a sort σ partitioned by two subsorts τ_1 and τ_2, where some feature is declared for both τ_1 and τ_2 but not for σ.

of a part of a linguistic entity. [8] Likewise, SRL as it stands is sufficiently expressive to encode some of the familiar ways of conceptualizing lexical rules in HPSG, such as the 'description-level' lexical rules of Meurers and Minnen 1998, and the closely related lexical-rules-as-sorts approach of Bouma 1997 and Abeillé et al. 1998, [9] and the same thing is true of linear precedence rules (see Richter and Sailer 1995). [10]

5 From Models to SGC: an Informal Synopsis

In section 3, we suggested, as a first approximation, that the SGC of a grammar could be identified with the objects in a model of the grammar. On the other hand, in HPSG practice, the linguistic entities (words, phases, and their various kinds of parts) are usually represented by feature structures. But how do we get from objects in a model to feature structures? In fact, almost effortlessly. Temporarily postponing the technicalities, we can explicate the relationship as follows. A model of a grammar is an interpretation of the grammar's signature, and an interpretation is a unary partial algebra. And, given an object of the interpretation, we can look at the singly-generated subalgebra generated in the interpretation by that object (using the denotations of the features as operations). But it is intuitively clear that a singly generated algebra determines a unique feature structure in a natural way: just take the generating object itself to be the root of the feature structure, the objects in the subalgebra to be its nodes, and use the operations to define the feature structure's labelled transitions in the obvious way. [11] Thus, every interpretation determines a unique set of feature structures in a simple and natural way. As far as SGC is concerned, it is these feature structures, not the model itself, that is of interest.

But more needs to be said. Notice that there is nothing to prevent two objects in a model of a grammar (or, more generally, in an interpretation of a signature) from determining isomorphic feature structures.

[8] Alternatively, SRL can be extended to allow for (bounded) quantification and definite relations; see Richter in prep..

[9] However, 'meta-level' lexical rules along the lines of Calcagno 1995, which require taking the closure of a set of relations on descriptions, are beyond the expressive power of SRL.

[10] SRL is not able to express the notion of 'set values' along the lines of Pollard and Moshier 1990. For an extension of SRL which is, see Alt 1996.

[11] To make the connection between objects and feature structures even more direct, we will simply *define* feature structures algebraically (instead of as graphs or as automata, as is usually done). In the terminology of the present paper, this works as follows: for a fixed signature $\mathbf{\Sigma} = \langle \mathsf{S}, \mathsf{F}, \mathsf{A} \rangle$, we define a ($\mathbf{\Sigma}$-)**feature structure** to be an ordered pair $\langle v, I \rangle$ where I is a $\mathbf{\Sigma}$-interpretation and v is an object which generates the universe of $\mathbf{\Sigma}$ in the sense that every object in that universe can be obtained by applying to v a function which is the denotation of a term.

On the other hand, as linguists, we are only interested in the feature structures in the set determined by the model *up to isomorphism*. As Pollard and Sag 1994 put it:

> "One thing that [language] certainly does not consist of is individual linguistic events or utterance tokens, for knowledge of these is not what is shared among the members of a linguistic community. Instead, what is known in common, that makes communication possible, is the system of linguistic types. For example, the type of the sentence *I'm sleepy* is part of that system, but no individual token of it is." (page 14)

The (abstract) feature structures

> "are intended to stand in a one-to-one relation with types of natural language expressions and their subparts. The role of the linguistic theory is to give a precise specification of which feature structures are to be considered admissible; the types of linguistic entities that correspond to the admissible feature structures constitute the predictions of the theory." (page 8)

Here Pollard and Sag are adopting an (unnecessary!) ontological commitment to types; the point of view of this paper is more adequately reflected by replacing 'type' with 'isomorphism class' in these passages. It follows from this that we should really only be interested in the isomorphism classes represented by the set of feature structures associated with the model. In fact, there is a well-known means of associating a unique feature structure—called an *abstract feature structure*—with each isomorphism class of feature structures, and moreover distinct abstract feature structures are associated with distinct isomorphism classes. (As throughout this section, we postpone the mathematical technicalities to the following section.) This means that we can associate with each object υ in an interpretation I a unique abstract feature structure $Abst_I(\upsilon)$, namely the abstract feature structure associated with the (isomorphism class) of the feature structure determined by υ. In this way, each interpretation determines a unique set of abstract feature structures. In case the interpretation happens to be a model of a given grammar, we can thus think of the set of abstract feature structures determined by the model as the SGC of the grammar.

But wait: not all models of a given grammar determine the same set of abstract feature structures! For example, it is not hard to see that, given any model of a grammar, any subalgebra of that model—even a singly generated one—of that model will also be a model; but clearly the subalgebra might determine a smaller set of abstract feature structures. And what if the model we have chosen as our touchstone of SGC is 'too small', in the sense that there is some object in some other model whose associated abstract feature structure is not among those determined by our chosen model?

There are a number of ways out of this fix, all of which lead to the same result. The first way is to forget about trying to chose a particular model to be be our touchstone for SGC. Instead, we form the set consisting of all those abstract feature structures which are $Abst_I(v)$ for some object v in some model of the grammar. This set, finally, is what we have been after: it is way of capturing the traditional notion of strong generative capacity. In fact, we might as well *define* the **sgc** of a grammar to be the set of abstract feature structures which are $Abst_I(v)$ for some object v in some interpretation I which is a model of the grammar.

Now consider the following question: among all the models of our grammar, can we find one which is 'so big' that the set of abstract feature structures determined by its objects includes the whole SGC of the grammar? As it happens, we can; in fact, as long as the grammar has any models at all, it will have a proper class of models with this property. We call such a model an **exhaustive** model. That is, a model is exhaustive just in case it contains, for any object v in any model, an object v' which determines the same abstract feature structure. This gives us a second route to SGC: we identify the SGC of a grammar with the set of abstract feature structures determined by some (any) exhaustive model.

A third way to obtain the SGC of a grammar (and the last we will consider here) is simpler than the others. We begin by limiting our attention to the set of (Σ-)abstract feature structures. We then select from it the subset of abstract feature structures which are models! What makes this so simple is the fact that we never need to construct exhaustive models, or consider proper classes of models. We only have to look at singly generated models, and, of those, only one from each isomorphism class.

To summarize, the SGC of an HPSG grammar is the set of its models which are abstract feature structures.

6 Technicalities

As before, we assume a fixed signature $\Sigma = \langle \mathsf{S}, \mathsf{F}, \mathsf{A} \rangle$. An interpetation of Σ is called **singly generated** if its universe contains an object v such that every object in the universe is **accessible** from v; here v' is defined to be **accessible** from v just in case it is the value at v of a function which is the denotation of a term. Now let $I = \langle U, S, F \rangle$ be an interpretation and υ one of its objects. We define the interpretation **generated by** υ in I, I_υ, to be $\langle U_\upsilon, S_\upsilon, F_\upsilon \rangle$, where

U_υ is the set of objects accessible from υ,

S_υ is the restriction of S to U_υ, and

F_υ is the total function from terms to partial functions from U_υ to U_υ whose value at each term τ is $F(\tau) \cap (U_\upsilon \times U_\upsilon)$.

Obviously, I_υ is a singly generated interpretation.

We define a **feature structure** to be an ordered pair $\langle \upsilon, I \rangle$ where I is an interpretation and υ is an object of I such that $I = I_\upsilon$. The objects of I are called the **nodes** of the feature structure, and υ, its **root**. For an arbitrary object υ in an interpretation I, we call $\langle \upsilon, I_\upsilon \rangle$ the feature structure **determined** by υ in I.

An **isomorphism** between two feature structures is just an algebraic isomorphism between their underlying interpretations which preserves the root. It is not hard to show that two feature structures are isomorphic just in case their respective roots have the same theory relative to their respective interpretations. [12]

We now single out a set of feature structures, called the **abstract** ones, which will serve as canonical representatives of isomorphism classes of feature structures. A couple of preliminary definitions are required. First, we define a set of terms β to be **prefix-closed** just in case

$: \in \beta$,

for each term τ and each feature φ, if $\tau\varphi \in \beta$ then $\tau \in \beta$.

Second, an equivalence relation ρ on β is called a **right congruence** just in case

for any two ρ-equivalent terms τ_1 and τ_2, for each feature φ,

if $\tau_1\varphi \in \beta$ then $\tau_1\varphi$ and $\tau_2\varphi$ are ρ-equivalent.

We now define a feature structure $\langle \upsilon, I \rangle$ to be **abstract** just in case the following conditions are satisfied:

There is a prefix-closed set of terms β and a right congruence ρ on β such that the nodes of the feature structure are the ρ-equivalence classes of β;

the root of the feature structure is the ρ-equivalence class of $:$;

for any two nodes υ, υ' and any feature φ, the function denoted by φ maps υ to υ' just in case there is a term τ in β such that

υ is the ρ-equivalence class of τ and

[12] Two objects (not necessarily in the same interpretation) are called **indiscernable** if they have the same theory relative to their respective interpretations. Thus two objects are indiscernible iff they determine isomorphic feature structures.

υ' is the ρ-equivalence class of $\tau\varphi$.

The utility of abstract feature structures is just that every feature structure is isomorphic to precisely one abstract feature structure. In fact, given an arbitrary feature structure $\langle \upsilon, I \rangle$, it is easy to see that there is a natural bijection from the objects of I to the set of ρ-equivalence classes of β, where β is the set of terms which denote functions defined at υ, and where two terms in β are ρ-equivalent just in case the functions they denote (are defined at and) have the same value at υ; the bijection in question maps each object to the equivalence class of all terms whose denotations map the root to that object. We then take these equivalence classes to be the nodes of an abstract feature structure, using the bijection to transfer the denotations of species and features from the orginal node set to the new node set in the obvious way. It is not hard to see that this construction reduces to the identity if the original feature structure is itself abstract; and two feature structures are associated in this way with the same abstract feature structure just in case they are isomorphic.

Now let I be an interpretation. We define a function $Abst_I$ from the objects of I to abstract feature structures in the obvious way: $Abst_I(\upsilon)$ is the abstract feature structure isomorphic to the feature structure determined by υ in I. In addition, we define the **abstract** of the interpretation, $A(I)$, to be the set whose members are the abstract feature structures $Abst_I(\upsilon)$ for υ an object of I.

Finally, let $\langle \Sigma, \theta \rangle$ be a fixed grammar. We say the grammar **generates** an abstract feature structure $\langle \upsilon, I \rangle$ just in case I is a model of the grammar. Obviously $\langle \upsilon, I \rangle$ is generated by the grammar just in case there is a model I' of the grammar such that $\langle \upsilon, I \rangle$ is in $A(I')$: in fact we can choose I' to be I itself, since $Abst_I(\upsilon) = \langle \upsilon, Inte \rangle$! It can also be shown that the grammar has an **exhaustive** model, namely a model I such that $A(I)$ is the SGC of the grammar. One such model can be constructed as follows: we take as its objects the set of pairs $\langle \upsilon', \langle \upsilon, I \rangle \rangle$ where $\langle \upsilon, I \rangle$ is a prediction of the grammar and υ' is one of its nodes. We leave it as an exercise for the reader to guess what the denotations of the species and features should be in this model.

References

Abeillé, Anne, Danièle Godard, and Ivan A. Sag. 1998. Two Kinds of Composition in French Complex Predicates. In Erhard Hinrichs, Andreas Kathol, and Tsuneko Nakazawa, eds., *Complex Predicates in Nonderivational Syntax.* Syntax and Semantics, vol. 30. New York: Academic Press.

Aït-Kaci, Hassan. 1984. *A Lattice-Theoretic Approach to Computation Based on a Calculus of Partially Ordered Type Structures.* Ph.D. thesis, University of Pennsylvania.

Aldag, Bjørn. 1997 *A Proof Theoretic Investigation of Prediction in HPSG.* Master's thesis, Eberhard-Karls-Universität, Tübingen.

Alt, Natali. 1996. *A Typed Feature Logic with Set-Valued Attributes as a Foundation for LP Rules.*

Bouma, Gosse. 1997. Monotonic Lexical Rules. Unpublished manuscript, Rijksuniversiteit Groningen.

Calcagno, Mike. 1995. Interpreting Lexical Rules. Paper presented at the Conference on Formal Grammar, Barcelona, July 1995.

Carpenter, Bob. 1992. *The Logic of Typed Feature Structures.* Cambridge Tracts in Theoretical Computer Science, No. 32. Cambridge: Cambridge University Press.

Gerdemann, Dale, and Paul J. King. 1994 The Correct and Efficient Implementation of Appropriateness Specifications for Typed Feature Structures. In *Proceedings of COLING 1994*, 956-960.

Johnson, Mark. 1988. *Attribute-Value Logic and the Theory of Grammar.* CSLI Lecture Notes, No. 14. Stanford: CSLI Publications.

Johnstone, Peter T. 1982. *Stone Spaces.* Cambridge Studies in Advanced Mathematics, No. 3. Cambridge: Cambridge University Press.

Kasper, Robert T., and William C. Rounds. 1986. A Logical Semantics for Feature Structures. In *Proceedings of the 24th Annual Conference of the ACM*, 235–242. New York.

Kepser, Stephan. 1994. A Satisfiability Algorithm for a Typed Feature Logic. Bericht Nr. 60, Arbeitspapiere des Sonderforschungsbereichs 340, Sprachtheoretische Grundlagen für die Computerlinguistik. Eberhart-Karls Universität, Tübingen.

King, Paul J. 1989. *A Logical Formalism for Head-Driven Phrase Structure Grammar.* Doctoral dissertation, University of Manchester, Manchester.

King, Paul J. In preparation. *Truth and Prediction in Head-Driven Phrase Structure Grammar.*

King, Paul J., and Kiril I. Simov. To appear. The Automatic Deduction of Classificatory Systems from Linguistic Theories. In *Grammars.*

King, Paul J., Kiril I. Simov, and Bjørn Aldag. To appear. The Complexity of Modellability in Finite and Computable Signatures of a Constraint Logic for Head-Driven Phrase Structure Grammar. In *Journal of Language, Logic, and Information.*

Meurers, W. Detmar, and Guido Minnen. 1998. A Computational Treatment of Lexical Rules in HPSG as Covariation in Lexical Entries. *Computational Linguistics.*

Moshier, M. Drew. 1988. *Extensions to Unification Grammars for the Description of Programming Languages.* Doctoral dissertation, University of Michigan, Ann Arbor.

Moshier, M. Drew, and William C. Rounds. 1987. A Logic for Partially Specified Data Structures. In *Proceedings of the 14th ACM Symposium on Principles of Programming Languages*, 156–167. Munich.

Pereira, Fernando, and Stuart Shieber. 1984. The Semantics of Grammar Formalisms Seen as Computer Languages. In *Proceedings of COLING 1984*, 123-129.

Pollard, Carl J. To appear. Sorts in Unification-Based Grammar and What they Mean. In Carl J. Pollard, *Lectures on Constraint-Based Grammar* Stanford: CSLI Publications.

Pollard, Carl J., and M. Drew Moshier. 1990. Unifying Partial Descriptions of Sets. In Philip P. Hanson, ed., *Information, Language, and Cognition.* Vancouver Studies in Cognitive Science, No. 1. Vancouver: University of British Columbia Press, 285-322.

Pollard, Carl J., and Ivan A. Sag. 1987. *Information-based Syntax and Semantics.* CSLI Lecture Notes, No. 13. Stanford: CSLI Publications.

Pollard, Carl J., and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar.* Chicago: Chicago University Press.

Richter, Frank, and Manfred Sailer. 1995. Remarks on Linearization: Reflections on the Treatment of LP-rules in HPSG in a Typed Feature Logic. Master's thesis, Eberhard-Karls-Universität, Tübingen.

Richter, Frank. In preparation. *Eine formale Sprache für HPSG und ihre Anwendung in einem Syntaxfrgment des Deutschen.* Ph.D. thesis, Eberhard-Karls-Universität, Tübingen.

Smolka, Gert. 1988. A Feature Logic with Subsorts. LILOG Report 33. Stuttgart: IBM Deutschland GmbH.