Arbeitspapiere des SFB 340
Bericht Nr. 00, Xxxx 1999

TOWARDS TRUTH IN HEAD-DRIVEN PHRASE STRUCTURE GRAMMAR

PAUL JOHN KING

Any linguistic paradigm with scientific aspirations must provide substantial answers to two questions. What does a grammar of a natural language claim of the natural language? And what qualifies as corroborating or confuting evidence for that claim? I have striven for longer than I am willing to casually admit to furnish the HPSG (Head-driven Phrase Structure Grammar) linguistic paradigm of (Pollard and Sag, 1994) with material responses to both questions, and I have felt for some years now that I have finally arrived at partial yet significant characterisations of two important notions:

truth, what it is for a grammar of a natural language to be true, and

verification, how the truth of a grammar of a natural language is empirically assessed.

All HPSGians ought to have an interest in good HPSG characterisations of truth and verification, since HPSG affords little of scientific consequence without them. Here I present only my characterisation of truth, and leave my characterisation of verification to a later paper. But this is only fitting, since my characterisation of truth in very large part informs my characterisation of verification.

I have taken great pains to present my characterisation in a manner that is engaging enough to attract and inform a broad HPSG readership. To this end, I have omitted all mathematical proofs and included only those mathematical definitions, propositions, theorems and suchlike that are indispensable to understanding the paper. I am sure that the consequent reduction in the intervention of mathematical technicalities between the reader and that which I most wish to convey to the reader will more than compensate for the undoubted mathematical cheapening

Received by the editors 10th March 1999.

ISSN 0947-6954/99

of the paper. However, the reader should not be lulled into thinking that this paper is easy to read. Far from it. Truth is a profound and subtle matter, and a woolly characterisation of truth offers merely a utility that is severely curtailed and a comprehensibility that is largely illusory. My characterisation is anything but woolly. It is rigorous enough to avoid omission, imprecision and contradiction. But such rigour demands much not only of the author but also of the reader. This paper is not light bedtime reading.

1. Introduction

Characterising truth in the HPSG of (Pollard and Sag, 1994) is far more difficult than most HPSGians realise, and a critical review of 1994 vintage HPSG is necessary before embarking upon a characterisation proper.

A notable hallmark of (Pollard and Sag, 1994) is the indispensable role it confers upon mathematical models in interpreting theories.

> In any mathematical theory about an empirical domain, the phenomena of interest are *modelled* by mathematical structures, certain aspects of which are conventionally understood as corresponding to observables of the domain. The theory itself does not talk directly about the empirical phenomena; instead, it talks about, or is *interpreted by*, the modelling structures. Thus 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 in natural language, say a technical dialect of English, German, or Japanese. But as theories become more complicated and their empirical consequences less straightforwardly apparent, the need for formalization arises. In cases of extreme formalization, of course, the empirical hypotheses are cast as a set of axioms in a logical language, where the modelling structures serve as the intended interpretations of expressions in the logic. (Pollard and Sag, 1994, page 6)

Armed with this methodological assumption, (Pollard and Sag, 1994) goes on to consider the relationship between a linguistic theory (or grammar) and a natural language.

> In our view, a linguistic theory should bear exactly the same relation to the empirical domain of natural language, namely, the universe of possible linguistic objects, as a mathematical theory of celestial mechanics bears to the possible motions of n-body systems. Thus we insist on being explicit as to what sorts of constructs are

assumed (i.e. what ontological categories of linguistic objects we suppose to populate the empirical domain) and on being mathematically rigorous as to what structures are used to model them. Moreover, we require that the theory itself actually count as a theory in the technical sense of precisely characterizing those modelling structures that are regarded as admissible or well-formed (i.e. corresponding to those imaginable linguistic objects that are actually predicted to be possible ones). This does not mean that the empirical hypotheses must be rendered in a formal logic as long as their content can be made clear and unambiguous in natural language (the same holds true in mathematical physics), but in principle they must be capable of being so rendered. Unless these criteria are satisfied, an enterprise purporting to be a theory cannot have any determinate empirical consequences. Thus we emphatically reject the currently widespread view which holds that linguistic theory need not be formalized. Rather, our position is the same as the one advocated by Chomsky (1957: 5):

> Precisely constructed models for linguistic structure can play an important role, both negative and positive, in the process of discovery itself. By pushing a precise but inadequate formulation to an unacceptable conclusion, we can often expose the exact source of this inadequacy and, consequently, gain a deeper understanding of the linguistic data. More positively, a formalized theory may automatically provide solutions for many problems other than those for which it was explicitly designed. Obscure and intuition-bound notions can neither lead to absurd conclusions nor provide new and correct ones, and hence they fail to be useful in two important respects. I think that some of those linguists who have questioned the value of precise and technical development of linguistic theory have failed to recognize the productive potential in the method of rigorously stating a proposed theory and applying it strictly to linguistic material with no attempt to avoid unacceptable conclusions by *ad hoc* adjustments or loose formulation.

In HPSG, the modelling domain—the analog of the physicist's flows—is a system of *sorted feature structures* (Moshier 1988; Pollard and Moshier 1990[1]), that 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

[1]I cite this as (Pollard and Moshier, 1994).

> types of linguistic entities that correspond to the admissible feature structures constitute the predictions of the theory.
>
> Just as in other empirical domains, linguistic theory has become sufficiently modular, complex, and deductive that a need for formalization has become apparent, especially to researchers concerned with the computational implementation of current theories. Thus in the past few years, a number of specialized 'feature logics' have been proposed (Kasper and Rounds 1986; Johnson 1988, 1991; Gazdar et al. 1988) for specifying constraints on the feature structures used in linguistic analysis. At the same time, theoretical computer scientists have proposed various constraint languages for programming language description (e.g. Moshier 1988) and knowledge representation (e.g. Höhfeld and Smolka 1988; Ait-Kaci and Nasr 1986) that are easily adapted to this end. A very recent integration of these lines of work is the development of feature logics appropriate for the formalization of linguistic theories, languages whose formulas serve as the linguist's analog of the space physicist's differential equations (see Carpenter 1990; Carpenter et al. 1991; Carpenter and Pollard 1991; King 1989; Pollard, n.d.[2]; Pollard and Carpenter 1990; and Carpenter 1992). The last-mentioned work in particular sets forth a logic very close to the one that we assume will underlie a fully formalized version of our theory. In very general terms, this can be characterized as a sorted variant of Kasper and Rounds's (1986) logic augmented with path inequalities, definite relations, and set values. (Pollard and Sag, 1994, pages 7 and 8)

Truth appears entirely straightforward on this account. A grammar is a set of formulae drawn from an augmented Kasper and Rounds logic; a grammar deems a feature structure admissible or inadmissible; the feature structures stand in a one-to-one correspondence with the linguistic types; a grammar predicts a linguistic type iff (if, and only if) the grammar admits the feature structure that corresponds to the linguistic type; and a grammar of a natural language is true only if the grammar predicts all and only the linguistic types in the natural language. However, a deeper consideration of the nature of a linguistic type casts doubt upon this characterisation.

(Pollard and Sag, 1994) discusses linguistic types in two particularly revealing passages. In the first, (Pollard and Sag, 1994) claims to have

[2] I cite this as (Pollard, 1989).

> accepted the conventional wisdom that linguistic theory must account for linguistic knowledge (a recursively definable system of linguistic types) but not necessarily for processes by which that knowledge is brought to bear in the case of individual linguistic tokens. Indeed, we take it to be the central goal of linguistic theory to characterize what it is that every linguistically mature human being knows by virtue of being a linguistic creature, namely, universal grammar. And a theory of a particular language—a grammar—characterizes what linguistic knowledge (beyond universal grammar) is shared by the community of speakers of that language. Indeed, from the linguist's point of view, that is what the language is.
>
> But what does language consist of? One thing that it 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.
>
> Just what sorts of things these linguistic types are is another question. Indeed, the precise ontological status of linguistic types is the subject of a very long-standing debate among various schools of conceptualists (e.g. Ferdinand de Saussure, Noam Chomsky), who take them to be mental objects, and realists (e.g. Leonard Bloomfield, Jerrold Katz, Paul Postal, Jon Barwise), who consider them to belong to extramental reality. Thus 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). For our part, we doubt that the question of whether objects of knowledge are mental or extramental is an empirical one. Fortunately, as Rich Thomason has pointed out, a successful science does not have to have solved its foundational problems: the interminable philosophical debate over the meaning of quantum mechanics has failed to diminish its predictive power. Our concern in this book will be with the internal architecture of the system that the linguistic types form, not with that system's ultimate ontological status. (Pollard and Sag, 1994, page 14)

Thus, linguistic types are emphatically *not* linguistic tokens, though some relationship between the two nonetheless exists. This relationship is apparently the usual relationship between types and tokens: a token is an individual instance, a type is a class of tokens, and a type *abstracts* a token (equivalently, the token *instances* the type) iff the token is a member of the type.

The relationship between linguistic types and linguistic knowledge appears more intimate. Indeed, it seems that linguistic knowledge *is* a system of linguistic types. However, the second passage reveals a less simplistic relationship.

> Modelling types of conceivable linguistic entities as rooted labelled graphs of a special kind—totally well-typed, sort-resolved feature structures—we formulate universal grammar and grammars of particular languages as a system of constraints on those feature structures. Only those feature structures that satisfy the constraints are taken to model (types of) grammatically well-formed linguistic entities. The distinction between the system of constraints and the collection of linguistic entities that satisfies it can be viewed as corresponding both to Chomsky's (1986a[3]) distinction between *I-language* and *E-language* and to Saussure's ((1916) 1959) distinction between *langue* and *parole*. Though only the latter is directly observable, only the former can be embodied as a mental computational system shared by members of a linguistic community.
>
> (Pollard and Sag, 1994, pages 57 and 58)

Thus emerges a tripartite ontology in which linguistic tokens are individual instances of natural language use, linguistic types are classes of linguistic tokens, and linguistic knowledge is a mental faculty that admits only certain linguistic types.

That (Pollard and Sag, 1994) characterises truth entirely in terms of linguistic types calls the characterisation into question. Almost all linguistic tokens are partly empirical. Linguistic communication would be almost impossible otherwise. But to what extent are linguistic types empirical? Is it an empirical matter whether a linguistic type is part of a natural language? To recast this seemingly innocuous question in a more concrete and discomfiting form, suppose that t_1 and t_2 are distinct tokens in a natural language L, and that two HPSG grammars of L apportion the linguistic tokens of L into two collections of linguistic types such that the first grammar admits a linguistic type that contains both t_1 and t_2 but the second grammar admits no such linguistic type. Are there any empirical grounds to prefer one grammar over the other?

[3]I cite this as (Chomsky, 1986).

The only possible basis for claiming that a linguistic type is part of a natural language is that the linguistic knowledge of an ordinary user of the natural language distinguishes that particular class of linguistic tokens. However, it is far from certain that linguistic knowledge *does* distinguish particular classes of linguistic tokens. And even if it does then the extent to which linguistic types are empirical cannot exceed the extent to which linguistic knowledge is empirical. Now, linguistic knowledge is certainly empirical in part. It has long been possible to surmise some linguistic knowledge from linguistic tokens, since linguistic knowledge must determine linguistic tokens to some degree. And recent technological advances make some mental processes directly observable. Nonetheless, the observation of linguistic knowledge itself is currently very limited and usually indirect.[4] Thus, (Pollard and Sag, 1994) literally characterises truth in such a manner that the veracity of a HPSG grammar of a natural language is immune – possibly in principle, and certainly in current practise – to experimental assessment. HPSGians must either content themselves with writing untestable theories and hoping for the appropriate advances in cognitive technology, or characterise truth anew in a fashion that is susceptible to currently available means of empirical appraisal. The latter course is clearly preferable, but how might it be pursued?

An obvious and appealing suggestion is to explicate the relationship between the linguistic types and the linguistic tokens in a natural language. A HPSG grammar of a natural language would then acquire empirical weight through this relationship just as, say, a theory of quantum chromodynamics acquires empirical weight through the relationship between the wholly metempirical quarks and gluons and the partly empirical hadrons, leptons and photons. Indeed, it could be argued that (Pollard and Sag, 1994) itself occasionally exploits some tacit such relationship. For example,[5] the English sentence "Adam thought Bill had crashed his car" has two readings, one in which Adam thought Bill had crashed Adam's car, and one in which Adam thought Bill had crashed Bill's car. For (Pollard and Sag, 1994), the distinction between these two readings rests upon the distinction between the linguistic types represented by feature structures F and G, where figure 1 very schematically depicts F, figure 2 depicts G, A and B are distinct but isomorphic substructures of F and G, and

[4]This is not a criticism of cognitive linguists but a respectful recognition of the magnitude of their undertaking.

[5]I assume throughout this example that the reader is familiar with the (Pollard and Sag, 1994) feature structures.

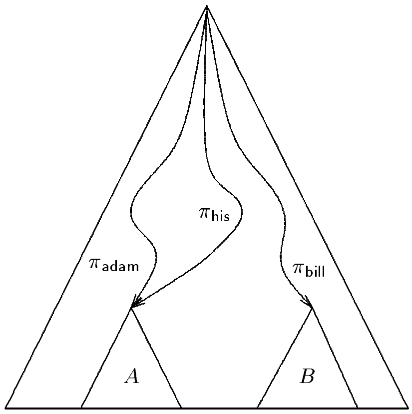
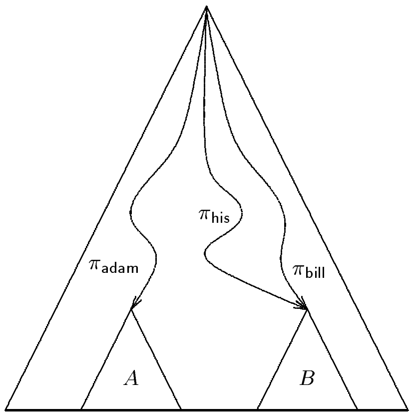


FIGURE 1. "Adam thought Bill had crashed his [Adam's] car"



,

FIGURE 2. "Adam thought Bill had crashed his [Bill's] car"

π_{adam}, π_{bill} and π_{his} are paths from the root nodes of F and G to A or B. The single difference between F and G is that π_{adam} and π_{his} re-enter – that is, lead to one and the same substructure – in F, whereas π_{bill} and π_{his} re-enter in G. What difference between the linguistic types represented by F and G does this difference between F and G signify? Since it is inconceivable that isomorphic feature structures such as A and B can represent distinct linguistic types, it cannot be that F represents the linguistic type in which π_{adam} and π_{his} lead to one and the same linguistic type while π_{bill} leads to a different linguistic type, whereas G represents the linguistic type in which π_{bill} and π_{his} lead to one and the same linguistic type while π_{adam} leads to a different linguistic type. Rather, F represents the linguistic type in *each token of* which π_{adam} and π_{his} lead to one and the same linguistic token while π_{bill} leads to a different linguistic token *of the same type*, whereas G represents the linguistic type in *each token of* which π_{bill} and π_{his} lead to one and the same linguistic token while π_{adam} leads to a different linguistic token *of the same type*. Re-entrancy in feature structures represents not identity of linguistic types but identity of linguistic tokens. On reflection, this assertion is uncontroversial, obvious even. Yet it unquestionably demonstrates that (Pollard and Sag, 1994) not only allows but actually uses an implicit relationship between linguistic types and linguistic tokens.

However, I have three objections to extending the (Pollard and Sag, 1994) characterisation of truth to incorporate an explicit relationship between linguistic types and linguistic tokens. Firstly, such an extension might violate the principle of ontological parsimony, a principle (Pollard and Sag, 1994) shares and nicely summarises as

> insofar as it is possible without doing violence to the simplicity and elegance of the theory, we do not posit constructs that do not correspond to observables of the empirical domain.
>
> (Pollard and Sag, 1994, page 9)

In light of the wholly metempirical nature of linguistic types, this principle entails that a characterisation of truth should avoid linguistic types unless to do so compromises the simplicity and elegance of HPSG grammars. Indeed, parsimony suggests that feature structures also be avoided where possible. Though (Pollard and Sag, 1994) insists that a theory of an empirical domain must be interpreted by a mathematical structure that models the domain, most scientists hold that the domain is itself a mathematical structure that directly interprets the theory. Thus, a theory of relativistic cosmology is interpreted not by a Riemannian geometry that *models* the space-time continuum, but rather by the Riemannian geometry that *is* the space-time continuum. There are numerous examples of well-known theories that are invariably interpreted directly. Euclid's axioms of geometry, Peano's axioms of arithmetic, Zermelo and Fraenkel's axioms of set theory: all are interpreted without the mediation of mathematical structures to model points, lines, planes, numbers or sets. Indeed, set theory completely undermines the (Pollard and Sag, 1994) position. What mathematical structure can model a set? In summary, much of the formal machinery that (Pollard and Sag, 1994) puts in place to characterise truth is superfluous and – all other things being equal – best dispensed with.

Secondly, the (Pollard and Sag, 1994) characterisation is poorly defined. Consider feature structures. (Pollard and Sag, 1994) is clear about some aspects of feature structures, but not others. We are unequivocally told that feature structures are both totally well-typed and sort resolved. But whether, to use the terminology of (Moshier, 1988), a feature structure is *concrete* – an individual graph – or *abstract* – an isomorphism class of graphs – is unclear. Both forms occur extensively in the feature-structure literature (Pollard and Sag, 1994) habitually cites. Indeed, both forms occur in (Pollard and Sag, 1987), the predecessor to (Pollard and Sag, 1994). To be sure, in a footnote discussing lexical entries, (Pollard and Sag, 1994, footnote, 61, page 53) says of a certain formula "(up to structural isomorphism) there is only one (totally well-typed, sort-resolved) feature structure that satisfies it." This strongly suggests that feature structures are concrete. Yet this same footnote mistakenly speaks of feature structures modelling linguistic tokens, and this is perhaps significant. While it is credible that two distinct linguistic tokens could be modelled by distinct but isomorphic concrete feature structures, it is beyond credibility that two distinct linguistic types could be so modelled. Also, (Pollard and Sag, 1994) nowhere states whether feature structures are necessarily finite or possibly infinite. Finite feature structures certainly predominate in the feature-structure literature (Pollard and Sag, 1994) cites, yet (Carpenter, 1992) clearly explicates the tensions that arise if feature structures are both totally well-typed and finite. In light of the equivocation or silence of (Pollard and Sag, 1994) on these matters, a full definition of a feature structure rests upon: an off-hand remark buried in a footnote that flatly contradicts the (Pollard and Sag, 1994) ontology; the readers intuitions about the nature of a linguistic type; the unstated preferences of (Pollard and Sag, 1994) within a diverse feature-structure literature; and the likelihood that (Pollard and Sag, 1994) recognises and resolves the tensions (Carpenter, 1992) discloses. This is anything but exact.

Or consider the formal language for admitting feature structures. (Pollard and Sag, 1994) defines no formal language of its own, but instead cites several feature logics and singles out the augmented Kasper and Rounds logic of (Carpenter, 1992) as "very close to the one that we assume will underlie a fully formalized version of our theory." A Kasper and Rounds logic uses feature structures to represent partial information, and (Carpenter, 1992) upholds this tradition.

> In this monograph we motivate and provide theoretical foundations for the specification and implementation of systems employing data structures which have come to be known as *feature structures*. Feature structures provide a record-like data structure for representing partial information that can be expressed in terms of features or attributes and their values. (Carpenter, 1992, page 1)

However, this concern with partial information calls into question the suitability of the logic of (Carpenter, 1992) for the HPSG of (Pollard and Sag, 1994). Though the earlier HPSG of (Pollard and Sag, 1987) uses feature structures as models of partial information about linguistic entities, the later HPSG of (Pollard and Sag, 1994) uses feature structures as models of the linguistic entities themselves.

> A common source of confusion is that feature structures themselves can be used as descriptions of other feature structures. Since feature structures that are not totally well-typed and sort-resolved can be arranged into a partial ordering relation (called the *subsumption* relation; see Shieber 1986 or P&S-87[6], Chapter 2), any feature structure can be thought of as partially describing any of the feature structures that it subsumes. Similarly, a feature structure can be taken as a partial description of any of the well-typed (or totally well-typed, or totally well-typed and sort-resolved) feature structures that it subsumes. We choose to eliminate this possible source of confusion by using only totally well-typed, sort-resolved feature structures as (total) models of linguistic entities and AVM diagrams (not feature structures) as descriptions.
> (Pollard and Sag, 1994, page 21)

This is no mere ontological quibble. In a Kasper and Rounds logic, many feature structures can satisfy a formula, but among the satisfiers is a so-called *most general* satisfier or set of satisfiers, and each formula is taken to indicate the partial information represented by its most general satisfier or set of satisfiers. (Moshier and Rounds, 1987) shows that if a Kasper and Rounds logic is augmented with unrestricted classical implication then some formulae have no most general satisfier or set of satisfiers, and so fail to indicate a piece of partial information. Consequently, every Kasper and Rounds logic with implication employs either a nonclassical

[6] I cite this as (Pollard and Sag, 1987).

implication as in (Moshier and Rounds, 1987) or a restricted classical implication as in (Carpenter, 1992). However, all of the constraints (Pollard and Sag, 1994) imposes on feature structures are classical implications, and some have a complexity that is difficult if not impossible to express within the strictures (Carpenter, 1992) imposes. This is not to criticise (Carpenter, 1992), which provides a logic perfectly adequate to the task it undertakes. But it is to criticise (Pollard and Sag, 1994), which specifically chooses from among the many feature logics it cites one that assumes feature structures model not linguistic entities but rather partial information about linguistic entities, and that expresses some HPSG constraints with difficulty if at all.

The two objections I have raised so far against extending the (Pollard and Sag, 1994) characterisation are not insurmountable: the ontological superfluities could surely be eliminated or indulged, and the definitional shortcomings rectified. Indeed, (Pollard, 1999) goes far to doing both. However, my third – and I believe insuperable – objection leads me to abandon any attempt to repair and extend the (Pollard and Sag, 1994) characterisation. A definition must be not only well-defined but also accurate, in the sense that the definition must faithfully and demonstrably capture the intuitive concept it is intended to express. To illustrate the importance of accuracy, consider a grammarian formulating within the (Pollard and Sag, 1994) HPSG paradigm a grammar of a natural language. The grammarian must construct a grammar that admits all and only the linguistic types in the natural language. Presumably, the grammarian has intuitions about linguistic types, and constructs their grammar in accord with those intuitions. However, the grammar is formally interpreted via the (Pollard and Sag, 1994) characterisation of truth. Thus, the grammar reflects the intuitions of the grammarian only to the extent that (i) the intuitions of the grammarian are commensurate with those that underlie the (Pollard and Sag, 1994) characterisation, and (ii) the (Pollard and Sag, 1994) formal machinery faithfully expresses those underlying intuitions. My gravest concern over the (Pollard and Sag, 1994) characterisation is that (Pollard and Sag, 1994) (and (Pollard, 1999)) barely mentions what those underlying intuitions are, far less demonstrates that the formal machinery faithfully expresses them. Nor can we rely on a consensual intuition shared generally by linguists. The fundamental ontological notion of the (Pollard and Sag, 1994) characterisation, the linguistic type, is an amorphous, artificial notion that linguists can and do construe in a variety of ways to suit their particular purposes. Witness the different ways members of the same collection of speech events are divided into types as the concerns of a linguist are phonetic or phonological. Rather, (Pollard and Sag, 1994) provides any grammarian wishing to use HPSG with a certain amount of formal machinery from which the grammarian must in effect attempt to surmise what intuitions inform the machinery and how well the machinery captures those intuitions. Not only is this an unreasonable, unpleasant and perverse imposition on the grammarian, but two factors severely hinder any grammarian in the attempt.

Firstly, I have already shown that (Pollard and Sag, 1994) is ambiguous, silent or contradictory about certain aspects of its formal machinery, and some of these aspects have a direct bearing on what intuitions the machinery expresses. For a somewhat contrived but nonetheless striking example, consider the infinite string Σ_∞, where

$$\Sigma_\infty = \text{"Adam thinks that Adam thinks that Adam thinks that} \ldots \text{"}.$$

Is Σ_∞ a grammatical English sentence? More to the point, does (Pollard and Sag, 1994) judge Σ_∞ grammatical? (Pollard and Sag, 1994) does not say whether infinite linguistic types are legitimate. Nor is there a consensus on the matter among linguists. Certainly, Σ_∞ could never be uttered or parsed, but many strings that could never be uttered or parsed are usually judged grammatical. After all, most grammarians – HPSGians included – hold that a grammar should express the knowledge of an ideal language user, and any English user can readily grasp Σ_∞. In a straw poll of my grammarian colleagues, roughly half claimed to have no reliable intuitions regarding the grammaticality of Σ_∞. Of the others, a small majority found Σ_∞ ungrammatical and a significant minority found Σ_∞ grammatical. Clearly, there is no consensus here. Of course, though (Pollard and Sag, 1994) says nothing *directly* about the legitimacy of infinite linguistic types, we do have recourse to the feature structures. If infinite feature structures are forbidden then Σ_∞ cannot be grammatical, but if infinite feature structures are allowed then I know of no existing HPSG grammar of English that deems grammatical each of

"Adam thinks"

"Adam thinks that Adam thinks"

"Adam thinks that Adam thinks that Adam thinks"

$\vdots$

yet deems Σ_∞ ungrammatical. Unfortunately, (Pollard and Sag, 1994) does not say whether feature structures must be necessarily finite or can be possibly infinite. Thus, whether the (Pollard and Sag, 1994) characterisation supports or rejects the intuition that Σ_∞ is grammatical rests upon a technical decision in the definition of a feature structure – whether to allow or forbid infinite feature structures – that must be taken with no guidance from (Pollard and Sag, 1994).

Infinite sentences are perhaps so bizarre that the above example could be dismissed as a mere curiosity, interesting but insignificant. However, the second hindering factor is, if anything, even more pernicious than the first. Simply put, those aspects of the (Pollard and Sag, 1994) formal machinery that *are* well-defined are formulated at too far a remove from intuitions regarding linguistic types for any grammarian to be sure that they have correctly divined the intuitions that inform even those aspects. For example, consider the stipulation that feature structures be totally well-typed. (Pollard and Sag, 1994) justify this stipulation by appealing to an intuition regarding the relationship between feature structures and linguistic entities. Any given grammarian might well share this intuition. That same grammarian might well also have the intuition that infinite linguistic types do not exist and feature structures must perforce be finite. However, (Carpenter, 1992) points out that requiring feature structures to be simultaneously finite and totally well-typed can have problematic consequences, and these consequences could easily conflict with other of the grammarian's intuitions. The source of this conflict need not be that the grammarian has inconsistent intuitions. Rather, the stipulation that all feature structures be sort-resolved and totally well-typed carries rather more intuitive 'baggage' than (Pollard and Sag, 1994) either realises or admits, 'baggage' that our hypothetical grammarian may well not bear. Unfortunately, discovering this extra 'baggage' is far from straightforward. Thus, even considering only those aspects of the (Pollard and Sag, 1994) formal machinery that *are* well-defined, it is nigh impossible for a grammarian to be certain of what intuitions they are implicitly agreeing to if they choose to use HPSG.

The foregoing objections cast serious doubt upon the utility, the desirability and even the possibility of extending the (Pollard and Sag, 1994) characterisation of truth to include linguistic tokens. Instead, I depart radically from the approach of (Pollard and Sag, 1994) and reformulate the characterisation afresh from first principles. Motivated by considerations of ontological simplicity, together with the conviction that much of what (Pollard and Sag, 1994) says concerning linguistic types is in fact better understood as concerning linguistic tokens, I give a construal of a natural language as a system of linguistic tokens. I then use this construal to formulate an alternative characterisation of truth with a parsimonious ontology, clearly stated intuitions and well-defined and demonstrably correct formal mechanisms.

2. Ontology

A complete account of the nature of a HPSG linguistic token would require a solution to one of the major open problems of philosophy, the nature of existence. Fortunately, my characterisation of truth needs nothing as elaborate as a full explication of tokenhood. It suffices that a token is an individual or particular. A token can be an object such as my computer, an event such as my typing these words at my computer keyboard, a state such as my being at work, a location such as the space occupied by my computer, or a complex such as the contents of the file system of my computer. There may well be other kinds of entity that should be called tokens, but this short list suffices for the present.

The notion of a *linguistic* token requires however two modifications to this naive concept. Firstly, the notion of a token usually implies the actual existence of the token. However, every mature English user would judge any of the infinitely many strings

"Adam thinks"

"Adam thinks that Adam thinks"

"Adam thinks that Adam thinks that Adam thinks"

$\vdots$

legitimate English sentences, though only finitely many of them could ever occur in English speech tokens. The intuition here is that a grammar of a natural language should account for the linguistic knowledge of mature users of the natural language, but not for the biological, physical and psychological limitations the universe imposes upon the exercise of that knowledge. I share this intuition, but maintain that a scientifically worthwhile account of linguistic knowledge can presently only be given largely in terms of linguistic behaviour. In order to reconcile this apparent antithesis, I endorse the notion of nonactual tokens – tokens that could exist in some universe but happen not to in ours – and assume that the linguistic knowledge of each mature user of a natural language determines which tokens – actual and nonactual – are tokens of the natural language. A grammar of a natural language is then true only if the grammar delimits the collection comprising all and only those tokens – actual and nonactual – that the linguistic knowledge of mature users of the natural language would deem tokens of the natural language. Note, my appeal to linguistic knowledge as an arbiter of the acceptability or otherwise of tokens avoids my criticisms of the (Pollard and Sag, 1994) characterisation of truth. I deny neither the existence of linguistic knowledge nor its role in determining linguistic behaviour. I deny only the ability to subject theoretical claims about linguistic knowledge to significant empirical examination other than by experimentally testing consequent claims about linguistic behaviour.

The second modification concerns idealisation. The linguistic knowledge of different mature users of a natural language will almost certainly not judge exactly the same tokens to be tokens of the natural language. Also, mature users of a natural language can, and often do, successfully participate in tokens of the natural language that, on measured reflection, all participants agree are not proper tokens of the natural language. As my notion of a token and construal of a natural language stand, a natural language is therefore a family of collections of tokens, and each collection contains improper tokens. This reduces linguistics to the study of ideolect and impropriety. To avoid this reduction, I appeal to the Chomskyan notion of an 'ideal speaker-hearer' or ideal user of a natural language, a fictional flawless representative of the entire community of mature users of the natural language. A token is then of a natural language iff the linguistic knowledge of an ideal user of the natural language deems it to be thus.

Concerning the HPSG specific anatomy of a linguistic token, two passages in particular from (Pollard and Sag, 1994) are illuminating. Firstly, (Pollard and Sag, 1994) points out that

> the feature structures employed in HPSG are *sorted*. This means simply that each node is labelled with a *sort symbol* that tells what type of object the structure is modelling; that is, there is one sort symbol for each basic type (ontological category) of construct .... 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*.
>
> (Pollard and Sag, 1994, pages 17 and 18)

Secondly, in a very detailed appendix, (Pollard and Sag, 1994) tells us that the final version of the English grammar employed in the first eight chapters of (Pollard and Sag, 1994)

> consists of a *sort hierarchy* and a set of *principles*. The sort hierarchy 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
>
> $$\sigma : \begin{bmatrix} \mathrm{F}_1\ \tau_1 \\ \ldots \\ \mathrm{F}_n\ \tau_n \end{bmatrix}$$
>
> 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 F_i is appropriate for all objects of sort σ, and (2) for any such object, the value of the F_i feature must be an object of sort τ_i.
>
> If sorts σ_1 and σ_2 bear 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. (Pollard and Sag, 1994, pages 395 and 396)

There is clearly some scope for interpreting these passages. It must also be borne in mind that (Pollard and Sag, 1994) takes a linguistic entity to be a linguistic type, not a linguistic token. Nonetheless, I strongly believe that (Pollard and Sag, 1994) is exploiting a tacit relationship between linguistic types and linguistic tokens in these passages. Indeed, the second passage immediately goes on to say

> Feature structures, *which serve as our mathematical models of token linguistic objects* [my emphasis], and which are the entities constrained by the grammar, are required to be *sort-resolved* and *totally well-typed*. (Pollard and Sag, 1994, page 396)

I therefore hold that these passages yield a reasonable reading that reveals much about the structure of a HPSG linguistic token.

The informal presentation in these passages of the algebraic properties of the sorts can be a little confusing. For example, one passage says "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*", suggesting that subsorts are higher than supersorts in the partial order of sorts. However, the other passage says "For two sorts σ and τ, τ is a *subsort* of σ if and only if it is dominated by σ", suggesting that subsorts are lower than supersorts in the partial order of sorts. Writing $\mathtt{Sort}$ for the set of sorts, and writing $\varsigma_1 \preceq \varsigma_2$ for sort ς_1 is a subsort of sort ς_2, the algebraic properties of the sorts can be unambiguously given as

$\mathtt{Sort}$ is a finite set,

for each $\varsigma \in \mathtt{Sort}$, $\varsigma \preceq \varsigma$,

for each $\varsigma_1 \in \mathtt{Sort}$, for each $\varsigma_2 \in \mathtt{Sort}$,

  if $\varsigma_1 \preceq \varsigma_2$ and $\varsigma_2 \preceq \varsigma_1$ then $\varsigma_1 = \varsigma_2$,

for each $\varsigma_1 \in \mathtt{Sort}$, for each $\varsigma_2 \in \mathtt{Sort}$, for each $\varsigma_3 \in \mathtt{Sort}$,

  if $\varsigma_1 \preceq \varsigma_2$ and $\varsigma_2 \preceq \varsigma_3$ then $\varsigma_1 \preceq \varsigma_3$,

for each $\varsigma_1 \in \mathtt{Sort}$, for each $\varsigma_2 \in \mathtt{Sort}$, for each $\varsigma_3 \in \mathtt{Sort}$,

  if $\varsigma_1 \preceq \varsigma_2$ and $\varsigma_1 \preceq \varsigma_3$ then $\varsigma_2 \preceq \varsigma_3$ or $\varsigma_3 \preceq \varsigma_2$,

object $\in \mathtt{Sort}$,

for each $\varsigma \in \mathtt{Sort}$, $\varsigma \preceq$ *object*,

for each $\varsigma_1 \in \mathtt{Sort}$, for each $\varsigma_2 \in \mathtt{Sort}$,

  ς_1 is a **daughter** of ς_2

  iff $\varsigma_1 \preceq \varsigma_2$,

    $\varsigma_1 \neq \varsigma_2$, and

    for each $\varsigma \in \mathtt{Sort}$,

      if $\varsigma_1 \preceq \varsigma$ and $\varsigma \preceq \varsigma_2$ then $\varsigma_1 = \varsigma$ or $\varsigma = \varsigma_2$, and

for each $\varsigma \in \mathtt{Sort}$,

  ς is **maximal** iff for each $\varsigma' \in \mathtt{Sort}$, if $\varsigma' \preceq \varsigma$ then $\varsigma' = \varsigma$.

I write $\mathtt{Daug}(\varsigma)$ for the set of daughters of sort ς.

The passages explain the semantic properties of the sorts less clearly than they explain the algebraic properties. Presumably, (Pollard and Sag, 1994) intends the terms "entities" and "objects" to refer to types. However, (Pollard and Sag, 1994) relies upon an implicit relationship between types and tokens in these locutions, and the terms "entities" and "objects" can more sensibly be taken to refer to tokens than to types.

Thus, sorts are symbols that denote collections of tokens. Writing `Univ` for the universe of linguistic tokens (the collection of "all linguistic entities with which the grammar deals"), and writing $[\![\varsigma]\!]$ for the collection of tokens that sort ς denotes, the semantic properties of the sorts can be formally expressed as

$[\![\mathit{object}]\!] = \mathtt{Univ}$, and

for each nonmaximal $\varsigma \in \mathtt{Sort}$,

for each $\varsigma' \in \mathtt{Daug}(\varsigma)$, for each $\upsilon \in [\![\varsigma']\!]$, $\upsilon \in [\![\varsigma]\!]$, and

for each $\upsilon \in [\![\varsigma]\!]$, for some unique $\varsigma' \in \mathtt{Daug}(\varsigma)$, $\upsilon \in [\![\varsigma']\!]$.

Features are also symbols, but they denote partial functions from `Univ` to `Univ`, and the feature declarations encode conditions on the domains and ranges of the denoted functions. Writing `Feat` for the set of features, and writing $[\![\varphi]\!]$ for the partial function that feature φ denotes, the feature declaration

$$\varsigma : \begin{bmatrix} \varphi_1 & \varsigma_1 \\ \vdots & \vdots \\ \varphi_n & \varsigma_n \end{bmatrix}$$

encodes the condition that

for each $i \in \{1, \ldots, n\}$, for each $\upsilon \in [\![\varsigma]\!]$,

$[\![\varphi_i]\!](\upsilon)$ is defined and $[\![\varphi_i]\!](\upsilon) \in [\![\varsigma_i]\!]$.

The algebraic properties of the features – namely "If sorts σ_1 and σ_2 bear declarations [F τ_1] and [F τ_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" and "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" – accord naturally with the semantic properties of the sorts and features.

Note that feature declaration

$$\varsigma : \begin{bmatrix} \varphi_1 & \varsigma_1 \\ \vdots & \vdots \\ \varphi_n & \varsigma_n \end{bmatrix}$$

encodes not the condition that

for each $\varphi \in \mathtt{Feat} \setminus \{\varphi_1, \ldots, \varphi_n\}$, for each $\upsilon \in [\![\varsigma]\!]$,

$[\![\varphi]\!](\upsilon)$ is undefined,

but rather the condition that

for each $\varphi \in \mathtt{Feat} \setminus \{\varphi_1, \ldots, \varphi_n\}$, for some $\upsilon \in [\![\varsigma]\!]$,

$[\![\varphi]\!](\upsilon)$ is undefined.

Thus, for each $\varsigma \in \mathtt{Sort}$, for each $\varphi \in \mathtt{Feat}$, there exist three possibilities:

$[\![\varphi]\!]$ is defined on all tokens in $[\![\varsigma]\!]$,

$[\![\varphi]\!]$ is defined on some but not all tokens in $[\![\varsigma]\!]$, and

$[\![\varphi]\!]$ is defined on no tokens in $[\![\varsigma]\!]$.

For example, in (Pollard and Sag, 1994),

$[\![$PHONOLOGY$]\!]$ is defined on all tokens in $[\![$*sign*$]\!]$,

$[\![$DAUGHTERS$]\!]$ is defined on some but not all tokens in $[\![$*sign*$]\!]$, and

$[\![$LOCAL$]\!]$ is defined on no tokens in $[\![$*sign*$]\!]$.

The feature declaration for ς distinguishes the first possibility from the second and third, but does not distinguish between the second and third possibilities. However, I read the statement "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" to mean that if ς is maximal then the second possibility is disallowed: for each maximal $\varsigma \in \mathtt{Sort}$, for each $\varphi \in \mathtt{Feat}$,

if ς bears a declaration of the form $[\varphi\, \varsigma']$
then $[\![\varphi]\!]$ is defined on each token in $[\![\varsigma]\!]$, and

if ς bears no declaration of the form $[\varphi\, \varsigma']$
then $[\![\varphi]\!]$ is defined on no token in $[\![\varsigma]\!]$.

3. Formalism

I mentioned earlier that (Pollard and Sag, 1994) envisages using an augmented Kasper and Rounds logic to formally express its HPSG grammars, but that such logics are poorly suited to the task since they assume that features structures represent partial information about entities, and moreover this assumption renders the expression in these logics of certain HPSG principles difficult and counterintuitive if not impossible. I therefore turn elsewhere for a formal language with which to express HPSG grammars.

The logics of (Johnson, 1988), (Smolka, 1988) and (King, 1989) all provide formal languages with unrestricted classical implication and formulae that denote sets of entities. All three fail in one respect to provide the formal language (Pollard and Sag, 1994) envisages: the formulae in each denote sets of entities, not sets of feature structure representations of entities. However, this may well be an ontological advantage. Of the three, I choose the SRL (Speciate Re-entrant Logic) of (King, 1989) because it is the only one of the three expressly designed for HPSG, and so contains several HPSG specific mechanisms the others do not. Though SRL is equipped with a sound, complete and decidable logic, I need and primarily discuss here only the syntax and semantics of SRL. Please note that I generalise some of the definitions and change some of the notation and terminology of (King, 1989) to better suit the present paper, and include a few notions that have entered SRL subsequent to (King, 1989).

Underlying a conventional interpretable logic is the intuition that an assertion is a finite and syntactically well-formed string of symbols that is either true or false when interpreted. Underlying SRL is the intuition that a description is a finite and syntactically well-formed string of symbols that is either true or false *of an entity* when interpreted. For example, the English sentence "it is black" is a description that is true of a soot particle but false of a snow flake when interpreted. To capture this intuition, SRL provides a class of formal languages. Each formal language is founded upon a signature and a class of interpretations. The signature provides the nonlogical symbols from which the descriptions are syntactically constructed. Each interpretation provides a universe of entities, and assigns each nonlogical symbol in the signature a meaning. A description is a finite and well-formed string of logical symbols and symbols from the signature, and a theory is a set of descriptions. Each interpretation determines a description denotation function and a theory denotation function. The former assigns each description the set of entities in the interpretation of which the description is true, and the latter assigns each theory the set of entities in the interpretation of which each description in the theory is true.

A **signature** is a triple $\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$ such that

S is a set,

F is a set, and

A is a total function from $\mathsf{S} \times \mathsf{F}$ to $Pow(\mathsf{S})$.

I call each member of S a **species** in Σ, each member of F a **feature** in Σ, and A the **appropriateness function** in Σ. For convenience, I henceforth assume that none of the symbols $:$, $\approx$, $\sim$, $\neg$, $\wedge$, $\vee$, $\rightarrow$, $[$ and $]$ is a species or a feature. For each signature $\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$, an **interpretation** of Σ is a triple $I = (U, S, F)$ such that

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

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

I call U the **universe** in I, and each member of U an **entity** in I. I sometimes write $\upsilon \in I$ for $\upsilon \in U$. I call I **trivial** iff U is empty. Each species denotes a set of entities such that the denotations of the species partition U. S assigns each entity the unique species that denotes the set to which the entity belongs. Thus, species σ denotes set $\{\upsilon \in U \mid S(\upsilon) = \sigma\}$ of entities. I write entity υ is **in** species σ to mean that $S(\upsilon) = \sigma$. Each feature denotes a partial function from U to U. F assigns each feature the partial function it denotes. I write feature φ is **defined** on entity υ and **maps** υ to entity υ' to mean that $F(\varphi)(\upsilon)$ is defined and $F(\varphi)(\upsilon) = \upsilon'$. The appropriateness function encodes – and the last clause in the definition of an interpretation enforces – a strict relationship between the denotations of species and features: if $\mathsf{A}(\sigma, \varphi) = \emptyset$ then feature φ is defined on no entity in species σ, but if $\mathsf{A}(\sigma, \varphi) \neq \emptyset$ then feature φ is defined on each entity in species σ and maps each entity in σ to an entity in some species in $\mathsf{A}(\sigma, \varphi)$.

For each signature $\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$, T_Σ and D_Σ 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 $\tau_1 \in \mathsf{T}_\Sigma$, for each $\tau_2 \in \mathsf{T}_\Sigma$, $\tau_1 \approx \tau_2 \in \mathsf{D}_\Sigma$,

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

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

I call each member of T_Σ a **term** in Σ, each member of D_Σ a **description** in Σ, and each subset of D_Σ a **theory** in Σ. For each signature $\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$, for each interpretation $I = (U, S, F)$ of Σ, T_I is the total function from T_Σ to the set of partial functions from U to U, D_I is the total function from D_Σ to $Pow(U)$, and Θ_I is the total function from $Pow(\mathsf{D}_\Sigma)$ to $Pow(U)$ such that

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

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

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

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 $\delta_2 \in \mathsf{D}_\Sigma$,

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

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

Each term denotes the composition of the denotations of its component features, applied in left to right order. T_I assigns each term the partial function it denotes. I write term τ is **defined** on entity υ and **maps** υ to entity υ' to mean that $T_I(\tau)(\upsilon)$ is defined and $T_I(\tau)(\upsilon) = \upsilon'$. A description is either true or false of an entity. Description $\tau_1 \approx \tau_2$ is true of entity υ iff τ_1 and τ_2 map υ to the same entity, description $\tau \sim \sigma$ is true of entity υ iff τ maps υ to an entity in σ, description $\neg\delta$ is true of entity υ iff δ is false of υ, description $[\delta_1 \wedge \delta_2]$ is true of entity υ iff δ_1 is true of υ and δ_2 is true of υ, description $[\delta_1 \vee \delta_2]$ is true of entity υ iff δ_1 is true of υ or δ_2 is true of υ (or both), and description $[\delta_1 \rightarrow \delta_2]$ is true of entity υ iff δ_2 is true of υ whenever δ_1 is true of υ. D_I assigns each description the set of entities of which the description is true. A theory is either true or false of an entity. Theory θ is true of entity υ iff each description in θ is true of υ. Θ_I assigns each theory the set of entities of which the theory is true.

For each signature Σ, for each interpretation I of Σ, for each $\theta \subseteq \mathsf{D}_\Sigma$,

I **satisfies** θ in Σ iff for some $\upsilon \in I$, $\upsilon \in \Theta_I(\theta)$, and

I **models** θ in Σ iff for each $\upsilon \in I$, $\upsilon \in \Theta_I(\theta)$.

Notice that, in SRL, an interpretation may satisfy a theory yet not model it, or may model a theory yet not satisfy it. For each signature Σ, for each $\theta \subseteq \mathsf{D}_\Sigma$, for each $\delta \in \mathsf{D}_\Sigma$,

θ **entails** δ in Σ iff for each interpretation I of Σ, $\Theta_I(\theta) \subseteq D_I(\delta)$.

For an extended example of the syntax and semantics of SRL at work in a simple and transparent setting, consider the following soap-operatic scenario:

> Anne and Adam are a married couple, as are Beth and Bill, but Cath and Carl are unmarried. Anne and Adam are best friends, as are Cath and Carl, but Beth's best friend is Carl and Bill's best friend is Anne.

I now construct one among many SRL formal languages that can generate and interpret terms, descriptions and theories about this scenario. Let

$\mathsf{S} = \{single, wife, husband\}$,

$\mathsf{F} = \{\text{SPOUSE}, \text{FRIEND}\}$,

A be the total function from $\mathsf{S} \times \mathsf{F}$ to $Pow(\mathsf{S})$ such that

$\mathsf{A}(single, \text{SPOUSE}) = \emptyset$,

$\mathsf{A}(single, \text{FRIEND}) = \{single, wife, husband\}$,

$\mathsf{A}(wife, \text{SPOUSE}) = \{husband\}$,

$\mathsf{A}(wife, \text{FRIEND}) = \{single, husband\}$,

$\mathsf{A}(husband, \text{SPOUSE}) = \{wife\}$, and

$\mathsf{A}(husband, \text{FRIEND}) = \{single, wife\}$, and

$\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$.

Σ is a signature. It provides the species *single*, *wife* and *husband*, the features SPOUSE and FRIEND, and encodes several restrictions on how these symbols may be interpreted. Informally stated, these restrictions are

no *single* has a SPOUSE,

each *single* has a FRIEND that is a *single*, a *wife* or a *husband*,

each *wife* has a SPOUSE that is a *husband*,

each *wife* has a FRIEND that is a *single* or a *husband*,

each *husband* has a SPOUSE that is a *wife*, and

each *husband* has a FRIEND that is a *single* or a *wife*.

However, while Σ encodes restrictions on how the species and features it provides may be interpreted, it does not itself provide an interpretation of those symbols. Rather, to enable the terms, descriptions and theories generated from Σ to refer to the scenario, the scenario must be construed as an interpretation of Σ. To begin, suppose that the entities in the interpretation are the people in the scenario. Therefore, let U be the set

$\{\text{Anne}, \text{Adam}, \text{Beth}, \text{Bill}, \text{Cath}, \text{Carl}\}$.

An entity in U could be in many different species – English person, married person, unemployed person, etc. – and nothing prevents the assignment to species *single*, *wife* and *husband* of any denotations whatsoever, provided the denotations partition U. Fortunately, if *single*, *wife* and *husband* are assigned their natural denotations – *single* denotes all the unmarried people in U, *wife* denotes all the married women in U, and *husband* denotes all the married men in U – then their denotations *do* partition U. Therefore, let S be the total function from U to S such that

$S(\text{Anne}) = \textit{wife}$,

$S(\text{Adam}) = \textit{husband}$,

$S(\text{Beth}) = \textit{wife}$,

$S(\text{Bill}) = \textit{husband}$,

$S(\text{Cath}) = \textit{single}$, and

$S(\text{Carl}) = \textit{single}$.

Equally, an entity in U could have many different features – mother, best friend, first lover, etc. – and features SPOUSE and FRIEND could be assigned any partial functions at all, provided the domains and ranges of those partial functions obey the restrictions encoded in Σ. Fortunately again, if SPOUSE and FRIEND are assigned their natural denotations – SPOUSE denotes the partial function that maps each person in U to their marital partner, and FRIEND denotes the partial function that maps each person in U to their best friend – then their denotations *do* obey the restrictions encoded in Σ. Therefore, let F be the total function from F to the set of partial functions from U to U such that

$F(\text{SPOUSE})$ is the partial function from U to U such that

$F(\text{SPOUSE})(\text{Anne}) = \text{Adam}$,

$F(\text{SPOUSE})(\text{Adam}) = \text{Anne}$,

$F(\text{SPOUSE})(\text{Beth}) = \text{Bill}$,

$F(\text{SPOUSE})(\text{Bill}) = \text{Beth}$,

$F(\text{SPOUSE})(\text{Cath})$ is undefined, and

$F(\text{SPOUSE})(\text{Carl})$ is undefined, and

$F(\text{FRIEND})$ is the partial function from U to U such that

$F(\text{FRIEND})(\text{Anne}) = \text{Adam}$,

$F(\text{FRIEND})(\text{Adam}) = \text{Anne}$,

$F(\text{FRIEND})(\text{Beth}) = \text{Carl}$,

$F(\text{FRIEND})(\text{Bill}) = \text{Anne}$,

$F(\text{FRIEND})(\text{Cath}) = \text{Carl}$, and

$F(\text{FRIEND})(\text{Carl}) = \text{Cath}$.

Let $I = (U, S, F)$. I is an interpretation of Σ, and terms, descriptions and theories concerning the characters in the scenario can now be generated from Σ and interpreted by I. For example,

terms

: is a term that maps Bill to Bill,

:SPOUSE is a term that maps Bill to Beth, since Beth is Bill's spouse,

:SPOUSE FRIEND is a term that maps Bill to Carl, since Carl is Beth's best friend, and

:SPOUSE FRIEND SPOUSE is a term that is undefined on Bill, since Carl has no spouse;

descriptions

:SPOUSE FRIEND $\sim$ *wife* is a description that is true only of Anne and Beth, since only Anne and Beth have spouses whose best friends are married women,

$\neg$:SPOUSE $\approx$:FRIEND is a description that is true only of Beth, Bill, Cath and Carl, since only Beth, Bill, Cath and Carl are not married to their best friends, and

[:SPOUSE FRIEND $\sim$ *wife* $\wedge$ $\neg$:SPOUSE $\approx$:FRIEND] is a description that is true only of Beth, since only Beth (i) has a spouse whose best friend is a married women, and (ii) is not married to her best friend; and

theories

{:SPOUSE FRIEND $\sim$ *wife*, $\neg$:SPOUSE $\approx$:FRIEND} is a theory that is true only of Beth, since only Beth (i) has a spouse whose best friend is a married women, and (ii) is not married to her best friend.

The last two examples illustrate a minor fact: a finite theory is semantically equivalent to the conjunction of its members. However, theories are not semantically redundant, since a theory can have infinitely many members whereas a conjunction can have only finitely many conjuncts. Finally, suppose that θ is the theory $\{: \sim \textit{wife}\}$. I certainly satisfies θ in Σ, since θ is true of Anne and Beth. However, I does not model θ in Σ, since θ is false of Adam, Bill, Cath and Carl. In fact, no nontrivial interpretation of Σ models θ in Σ. Suppose that I' is an interpretation of Σ, υ is an entity in I', and I' models θ in Σ. Then θ is true of υ. Thus, υ is in *wife*. Thus, feature SPOUSE is defined on υ and maps υ to an entity υ' in *husband*. Thus, θ is false of υ'. Thus, I' does not model θ in Σ. Since this contradicts the starting supposition, the starting supposition must be false. Hence, no nontrivial interpretation of Σ can model θ in Σ.

My characterisation of truth does not use the logic of SRL, so I do not present the logic in this paper. However, for those readers interested in logical issues, I give here a brief survey of the logical properties of SRL and references to lengthier expositions. Other readers can safely omit this paragraph. For each signature $\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$, I say that

Σ is **limited** iff S is finite,

Σ is **computable** iff S is finite, F is countable and A is recursive, and

Σ is **finite** iff S is finite and F is finite.

Notice that each finite signature is computable, and each computable signature is limited. (King, 1989) furnishes each limited signature with a Hilbert and Ackermann style calculus, and proves that the inference relation this calculus determines is sound and complete with respect to entailment: for each limited signature Σ, for each $\theta \subseteq \mathsf{D}_\Sigma$, for each $\delta \in \mathsf{D}_\Sigma$,

θ entails δ in Σ iff θ infers δ in Σ.

For each limited signature Σ, for each $\delta \in \Sigma$,

δ is a **theorem** in Σ iff $\emptyset$ infers δ in Σ.

An immediate consequence of the soundness and completeness of inference is that for each limited signature Σ, for each $\delta \in \Sigma$,

δ is a theorem in Σ iff no interpretation of Σ satisfies $\{\neg\delta\}$ in Σ.

(Kepser, 1994) shows that satisfiability is decidable in all computable signatures: for each computable signature Σ,

$\left\{\delta \in \mathsf{D}_\Sigma \,\middle|\, \text{some interpretation of } \Sigma \text{ satisfies } \{\delta\} \text{ in } \Sigma\right\}$ is total recursive.

Thus, SRL is decidable in all computable signatures: for each computable signature Σ,

$\left\{\delta \in \mathsf{D}_\Sigma \,\middle|\, \delta \text{ is a theorem in } \Sigma\right\}$ is total recursive.

However, (King et al., 1999) shows that nontrivial modellability is undecidable, even in some finite signatures: for some finite signature Σ,

$\left\{\delta \in \mathsf{D}_\Sigma \,\middle|\, \text{some nontrivial interpretation of } \Sigma \text{ models } \{\delta\} \text{ in } \Sigma\right\}$ is Π^0_1-complete.

Turning from logic to linguistics, recall that a HPSG grammar "consists of a *sort hierarchy* and a set of *principles*." A sort hierarchy can be expressed as a signature. I write $\mathtt{Maxi}(\varsigma)$ for the set of maximal subsorts of sort ς, that is,

$$\text{for each } \varsigma \in \mathtt{Sort},\ \mathtt{Maxi}(\varsigma) = \left\{\sigma \in \mathtt{Sort} \,\middle|\, \sigma \text{ is maximal and } \sigma \preceq \varsigma \right\}.$$

Let

$\mathsf{S} = \left\{\sigma \in \mathtt{Sort} \,\middle|\, \sigma \text{ is maximal} \right\}$,

$\mathsf{F} = \mathtt{Feat}$,

A be the total function from $\mathsf{S} \times \mathsf{F}$ to $Pow(\mathsf{S})$ such that

for each $\sigma \in \mathsf{S}$, for each $\varphi \in \mathsf{F}$,

$$\mathsf{A}(\sigma, \varphi) = \begin{cases} \mathtt{Maxi}(\varsigma) & \text{if } \sigma \text{ bears a declaration} \\ & \qquad\qquad \text{of the form } [\varphi\, \varsigma] \\ \emptyset & \text{otherwise} \end{cases}$$

$\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$,

$U = \mathtt{Univ}$,

S be the total function from U to S such that

for each $\upsilon \in U$, $\upsilon \in [\![S(\upsilon)]\!]$,

F be the total function from F to the set of partial functions from U to U such that

for each $\varphi \in \mathsf{F}$, $F(\varphi) = [\![\varphi]\!]$, and

$I = (U, S, F)$.

Σ is obviously a signature. Less obvious are two other facts. Firstly, I is an interpretation of Σ. Secondly, though a signature lacks much of the structure of a sort hierarchy – a signature has no notion of subsort for example – the only important information encoded in a sort hierarchy concerns the denotations of sorts and features, and all of this information can be easily inferred from the corresponding signature. Both facts follow from the observation that for each $\varsigma \in \mathtt{Sort}$,

for each $\sigma \in \mathtt{Maxi}(\varsigma)$, for each $\upsilon \in [\![\sigma]\!]$, $\upsilon \in [\![\varsigma]\!]$, and

for each $\upsilon \in [\![\varsigma]\!]$, for some unique $\sigma \in \mathtt{Maxi}(\varsigma)$, $\upsilon \in [\![\sigma]\!]$.

Firstly, $[\![\mathit{object}]\!] = \mathtt{Univ}$ and $\mathtt{Maxi}(\mathit{object}) = \mathsf{S}$. Thus, for each $\upsilon \in \mathtt{Univ}$, for some unique $\sigma \in \mathsf{S}$, $\upsilon \in [\![\sigma]\!]$. Thus, S is well-defined. Secondly, the set of entities denoted by a sort is fully determined by the sets of entities denoted by the maximal subsorts of the sort, and the domain and range of the partial function denoted by a feature is fully determined by the feature declarations of the maximal sorts. In short, the nonmaximal sorts of a sort hierarachy, their denotations, their feature declarations and their subsort relationships are superfluous. A signature can express the essential information encoded in a sort hierarchy without the superfluity. A nonmaximal sort becomes merely a symbol that represents a set of maximal sorts, and the only necessary addition to SRL is the following metasyntactic convention: for each $\tau \in \mathsf{T}_\Sigma$, for each nonmaximal $\varsigma \in$ `Sort`,

if ς represents the set $\{\sigma_1, \ldots, \sigma_n\}$ of maximal sorts
then I write $\tau \sim \varsigma$ for $[\tau \sim \sigma_1 \vee \ldots \vee \tau \sim \sigma_n]$.

Viewing nonmaximal sorts as symbols that represent sets of maximal sorts also confers another subtle advantage on a signature over a sort hierarchy. HPSGians occasionally use the notion of 'multiple inheritance' in which one sort is a daughter of two distinct mother sorts. In particular, a lexical hierarchy (see (Pollard and Sag, 1994, pages 36, 37 and 395)) is usually presented as a sort hierarchy with multiple inheritance. Unfortunately, the (Pollard and Sag, 1994) definition of a sort hierarchy doubly precludes multiple inheritance. Firstly, (Pollard and Sag, 1994) insists that "The sort hierarchy is presented as a taxonomic tree, with the root labelled *object* (the sort of all linguistic entities with which the grammar deals)." Secondly, even if this first stipulation is relaxed to allow multiple inheritance, (Pollard and Sag, 1994) also requires that "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." This prevents multiple inheritance in any meaningful sense, since any sort that is the daughter of two distinct mother sorts must perforce denote the empty set. Moreover, to relax this second requirement would be to seriously compromise other information encoded in a sort hierarchy regarding the domains and ranges of the denotations of features. A signature, on the other hand, can allow multiple inheritance while maintaining all of the feature denotation information encoded in a sort hierarchy. For example, introducing into our earlier example the sorts *married*, *non-wife*, *non-husband* and *person* – where *married* represents $\{\textit{wife}, \textit{husband}\}$, *non-wife* represents $\{\textit{single}, \textit{husband}\}$, *non-husband* represents $\{\textit{single}, \textit{wife}\}$ and *person* represents $\{\textit{single}, \textit{wife}, \textit{husband}\}$ – in no way weakens the information encoded in the signature concerning the domains and ranges of features SPOUSE and FRIEND. Yet sort *single* has distinct mother sorts *non-wife* and *non-husband* as well as a nonempty denotation.

A set of principles can be expressed as a theory. This is best illustrated with examples.[7] A simple example is one of the most important of the (Pollard and Sag, 1994) principles, the 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
>
> (Pollard and Sag, 1994, page 399)

where "a *headed phrase* is a *phrase* whose DAUGHTERS value is of sort *headed-structure*." This principle can be simply and directly expressed in SRL as

$$\left[\begin{array}{l} \left[\ :\sim \textit{phrase} \ \wedge\ \text{:DAUGHTERS} \sim \textit{headed-structure}\ \right] \\ \rightarrow \quad \text{:SYNSEM LOCAL CATEGORY HEAD} \approx \text{:DAUGHTERS} \\ \qquad\qquad \text{HEAD-DAUGHTER SYNSEM LOCAL CATEGORY HEAD} \end{array}\right].$$

A more complicated example is the equally important subcategorization principle:

> In a headed phrase, the list value of DAUGHTERS | HEAD-DAUGHTER | SYNSEM | LOCAL | CATEGORY | SUBCAT is the concatenation of the list value of SYNSEM | LOCAL | CATEGORY | SUBCAT with the list consisting of the SYNSEM values (in order) of the elements of the list value of DAUGHTERS | COMPLEMENT-DAUGHTERS.
>
> (Pollard and Sag, 1994, page 399)

One could imagine expressing the subcategorization principle in SRL as

$$\left[\begin{array}{l} \left[\ :\sim \textit{phrase} \ \wedge\ \text{:DAUGHTERS} \sim \textit{headed-structure}\ \right] \\ \rightarrow \texttt{concat} \left(\begin{array}{l} \text{:DAUGHTERS HEAD-DAUGHTER SYNSEM LOCAL} \\ \qquad \text{CATEGORY SUBCAT}, \\ \text{:SYNSEM LOCAL CATEGORY SUBCAT}, \\ \texttt{extract}(\text{:DAUGHTERS COMPLEMENT-DAUGHTERS}) \end{array}\right) \end{array}\right],$$

where `extract` is a function that maps each list of *sign*s to a list comprising the SYNSEM values (in order) of the members of the list of *sign*s, and `concat` is a relation that holds of three lists iff the first list is the concatenation of the second and the third. However, this principle seems to be beyond the expressive ambit of SRL, since its expression seems to require a function and a relation that are neither sorts nor features. Indeed, functions and relations such as `extract` and `concat` strictly lie outside the syntax and semantics of *any* feature logic. To deal with such functions and relations, a feature logic is typically augmented with a small repertoire of function and relation symbols that can be embedded within

[7] I assume throughout this illustration that the reader is familiar with the (Pollard and Sag, 1994) principles.

the formulae proper of the feature logic. The interpretation of these additional symbols is fixed independently of the feature logic, and any model- or proof-theoretic properties of the feature logic do not necessarily carry over to the augmented language. However, it is perhaps surprising that SRL *can* express the subcategorization principle without augmentation, albeit inelegantly, via a so-called 'junk slot' encoding. Some readers might wish to omit the following paragraph.

A junk slot encoding uses 'junk' features to construct entities with recursive structures. Suppose that $\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$ is the signature corresponding to the (Pollard and Sag, 1994) sort hierarchy. Then each of the sorts *empty-list*, *nonempty-list*(*sign*), *nonempty-list*(*synsem*) and *phrase* is a species in Σ. Let

$\mathsf{S}' = \mathsf{S} \cup \{junk, stop\}$,

$\mathsf{F}' = \mathsf{F} \cup \{\text{JUNK}, \text{RESULT}, \text{LEFT}, \text{RIGHT}\}$,

A' extend A such that

$\mathsf{A}(phrase, \text{JUNK}) = \{junk\}$,

$\mathsf{A}(junk, \text{JUNK}) = \{junk, stop\}$,

$\mathsf{A}(junk, \text{RESULT}) = \{\textit{empty-list}, \textit{nonempty-list}(synsem)\}$,

$\mathsf{A}(junk, \text{LEFT}) = \{\textit{empty-list}, \textit{nonempty-list}(synsem)\}$,

$\mathsf{A}(junk, \text{RIGHT}) = \{\textit{empty-list}, \textit{nonempty-list}(sign)\}$, and

$\mathsf{A}(\sigma, \varphi) = \emptyset$ for all other $(\sigma, \varphi) \in (\mathsf{S}' \times \mathsf{F}') \setminus (\mathsf{S} \times \mathsf{F})$, and

$\Sigma' = (\mathsf{S}', \mathsf{F}', \mathsf{A}')$.

Σ' is a signature. Let

$$\delta_1 = \left[\begin{array}{l} \left[\ :\sim phrase \ \wedge \ \text{:DAUGHTERS} \sim \textit{headed-structure}\ \right] \\ \rightarrow \left[\begin{array}{l} \text{:JUNK} \sim junk \wedge \\ \text{:JUNK RESULT} \approx \text{:DAUGHTERS HEAD-DAUGHTER} \\ \qquad \text{SYNSEM LOCAL CATEGORY SUBCAT} \wedge \\ \text{:JUNK LEFT} \approx \text{:SYNSEM LOCAL CATEGORY SUBCAT} \wedge \\ \text{:JUNK RIGHT} \approx \text{:DAUGHTERS} \\ \qquad \text{COMPLEMENT-DAUGHTERS} \end{array}\right] \end{array}\right],$$

$$\delta_2 = \left[\begin{array}{l} \left[\ :\sim junk \ \wedge \ \text{:LEFT} \sim \textit{empty-list} \ \wedge \ \text{:RIGHT} \sim \textit{empty-list}\ \right] \\ \rightarrow \left[\begin{array}{l} \text{:JUNK} \sim stop \wedge \\ \text{:RESULT} \approx \text{:RIGHT} \end{array}\right] \end{array}\right],$$

$$\delta_3 = \left[\begin{array}{l} \left[\begin{array}{l} : \sim \textit{junk} \ \land \ \text{:LEFT} \sim \textit{empty-list} \\ \land \text{:RIGHT} \sim \textit{nonempty-list}(\textit{sign}) \end{array} \right] \\ \rightarrow \left[\begin{array}{l} \text{:JUNK} \sim \textit{junk} \land \\ \text{:JUNK RESULT} \approx \text{:RESULT REST} \land \\ \text{:JUNK LEFT} \approx \text{:LEFT} \land \\ \text{:JUNK RIGHT} \approx \text{:RIGHT REST} \land \\ \text{:RESULT} \sim \textit{nonempty-list}(\textit{synsem}) \land \\ \text{:RESULT FIRST} \approx \text{:RIGHT FIRST SYNSEM} \end{array} \right] \end{array} \right], \text{ and}$$

$$\delta_4 = \left[\begin{array}{l} \left[\ : \sim \textit{junk} \ \land \ \text{:LEFT} \sim \textit{nonempty-list}(\textit{synsem}) \ \right] \\ \rightarrow \left[\begin{array}{l} \text{:JUNK} \sim \textit{junk} \land \\ \text{:JUNK RESULT} \approx \text{:RESULT REST} \land \\ \text{:JUNK LEFT} \approx \text{:LEFT REST} \land \\ \text{:JUNK RIGHT} \approx \text{:RIGHT} \land \\ \text{:RESULT} \sim \textit{nonempty-list}(\textit{synsem}) \land \\ \text{:RESULT FIRST} \approx \text{:LEFT FIRST} \end{array} \right] \end{array} \right].$$

A little effort shows that for each entity υ' in *junk*,

if LEFT maps υ' to a list $(x_1, \ldots, x_m)$ of entities in *synsem*,

RIGHT maps υ' to a list $(y_1, \ldots, y_n)$ of entities in *sign*,

for each $i \in \{1, \ldots, n\}$, SYNSEM maps y_i to an entity z_i in *synsem*, and

δ_2, δ_3 and δ_4 are true of υ'

then RESULT maps υ' to the list $(x_1, \ldots, x_m, z_1, \ldots, z_n)$ of entities in *synsem*.

Thus, for each headed phrase υ,

if JUNK maps υ to an entity υ' in *junk*,

:SYNSEM LOCAL CATEGORY SUBCAT maps υ to a list $(x_1, \ldots, x_m)$ of entities in *synsem*,

:DAUGHTERS COMPLEMENT-DAUGHTERS maps υ to a list $(y_1, \ldots, y_n)$ of entities in *sign*,

for each $i \in \{1, \ldots, n\}$, SYNSEM maps y_i to an entity z_i in *synsem*,

δ_1 is true of υ, and

δ_2, δ_3 and δ_4 are true of υ'

then :DAUGHTERS HEAD-DAUGHTER SYNSEM LOCAL CATEGORY SUBCAT maps υ to the list $(x_1, \ldots, x_m, z_1, \ldots, z_n)$ of entities in *synsem*.

Thus, theory $\{\delta_1, \delta_2, \delta_3, \delta_4\}$ expresses the subcategorization principle.[8]

[8]While this example demonstrates that SRL *can* express the subcategorization principle and similar using junk slot encodings, it also demonstrates abundantly that such

By means of junk slot encodings and other technical devices, a theory can express most, if not all, of the (Pollard and Sag, 1994) principles. Additionally, SRL can express or emulate many aspects of a HPSG grammar that are formulated at best loosely in (Pollard and Sag, 1994) and elsewhere in the HPSG literature. I briefly mention here the four examples I am most familiar with. Firstly, many HPSGians maintain that the construction of a lexicon requires the construction of a lexical hierarchy, a florid sort hierarchy comprising lexical sorts and equipped with multiple inheritance. I think this approach to the lexicon is misguided. The supporters of the approach point out that (Pollard and Sag, 1994) claims the logic of (Carpenter, 1992) to be "very close to the one that we assume will underlie a fully formalized version of our theory", and that the restricted classical implication of (Carpenter, 1992) necessitates a lexical hierarchy in order to formulate constraints on words. However, I have already voiced my contention that – notwithstanding the merits of the (Carpenter, 1992) logic – (Pollard and Sag, 1994) is wrong to base its HPSG on this particular logic. Nonetheless, I have shown above that a signature (unlike a sort hierarchy) can emulate multiple inheritance, thus making possible the construction of lexical hierarchies. Secondly, many HPSG accounts of the lexicon employ the informal notion of lexical rules, which the HPSG orthodoxy claims

> may be interpreted as rules of inference that derive lexical entries of inflected, derived, or compound words from those of simpler words
> (Pollard and Sag, 1994, page 37)

However, all attempts to formalise this notion yield little or nothing. (Meurers and Minnen, 1997) challenges this orthodoxy, and argues that the goals set for lexical rules can be better achieved by constraints that

---

encodings are extremely *ad hoc*, cumbersome and counterintuitive. Moreover, while every reasonable HPSG function and relation I have so far examined in detail can be expressed in SRL using junk-slot encodings, it is unclear exactly which functions and relations can be so encoded, which not, and whether those that can be encoded suffice for HPSG. Clearly, a SRL-like language equipped with a native mechanism for expressing functions and relations such as `extract` and `concat` would be a great improvement on SRL. To this end, Frank Richter and Manfred Sailer have conceived RSRL (Relational Speciate Re-entrant Language), an extension of the syntax and semantics of SRL with a perspicuous interpretation of relations and functions formulated specifically to meet the practical needs of the HPSGian. See (Richter, 1999) and (Richter et al., 1999) for details. However, the expressive benefits of RSRL over SRL exact a price. Unlike for SRL, there can be no sound and complete logic for RSRL. Nonetheless, many of the ideas presented in the present paper transfer *mutatis mutandis* to RSRL, and a number of RSRL counterparts of important results pertinent to the present paper have already been proved in ongoing work.

derive not lexical entries from lexical entries but rather lexical entities from lexical entities. These lexical constraints are then little different in kind from conventional principles, and as such can be readily expressed as theories. Thirdly, (Pollard and Sag, 1994) treats phonology very naively, being content to gloss phonological entities as lists of phoneme strings. (Höhle, 1999), on the other hand, presents a detailed SRL-compatible architecture for the expression of accounts of phonology. Moreover, the unavoidable interaction between phonology and the physical universe leads (Höhle, 1999) to propose an ontology of tokens that is far more sophisticated than mine. Among the highlights of this delightful ontology is a relationship between linguistic entities and possible physical events that is determinable solely by an adequately rich theory of the physical performance of phonological gestures. Finally, HPSG is one of a number of linguistic paradigms that adopts a looser relationship between constituent structure and word order than the Chomskyan paradigms usually adopt. Several papers propose dedicated list-handling mechanisms to constrain word order. These mechanisms are very much like `extract` and `concat`, in that they are extralogical additions to the syntax and semantics of an underlying feature logic. However, there is good evidence that SRL can express word-order constraints without such additional mechanisms. By extensive use of junk slots, (Richter and Sailer, 1995) formulates within SRL a broad, thorough and sophisticated account of word order in the German Mittelfeld, a domain known to be problematic for conventional approaches to specifying word order.

In summation, a HPSG grammar can be expressed in very large part, if not in entirety, as a SRL grammar, where a SRL **grammar** is a pair (Σ, θ) such that Σ is a signature and $\theta \subseteq \mathsf{D}_\Sigma$.

4. Truth

Equipped with an ontology and a formalism, I now characterise truth in HPSG by formulating a strong and well-defined necessary condition for a HPSG grammar to be true of a natural language. I proceed by presenting a succession of interim theses regarding truth in HPSG, finding fault and dispensing with each in turn until I arrive at a satisfactory thesis. Each thesis embodies a number of intuitions concerning truth in HPSG, and the process of rejecting and reformulating the theses can be seen as the process of criticising and refining those intuitions.

In section 1 of this paper, I argued for linguistic tokens as suitable media for characterising truth in HPSG. In section 2, I outlined a particular construal of a HPSG linguistic token based largely upon the structure a

(Pollard and Sag, 1994) sort hierarchy imposes upon such a token. In section 3, I demonstrated that a SRL signature more efficiently imposes upon a linguistic token the entirety of the structure a (Pollard and Sag, 1994) sort hierarchy imposes upon it. My first interim thesis encapsulates all of this work by formalising a simple intuition: a grammar is true of a natural language only if the natural language can be construed as a system of linguistic tokens meeting the conditions imposed upon them by the signature component of the grammar.

INTERIM THESIS 1. *For each grammar* $\Gamma = (\Sigma, \theta)$, *for each natural language* I,

if Γ *is true of* I

then I *is an interpretation of* Σ.

Unfortunately, under interim thesis 1, the theory component of a grammar plays no role in determining whether the grammar is true of a natural language. This is clearly absurd, and while I maintain that interim thesis 1 is true, I also concede that merely being an interpretation of a signature is far too weak a necessary condition for linguistic truth. To strengthen this condition, I observe that each (Pollard and Sag, 1994) linguistic principle is an implicational description, and for good reason. If a HPSGian wishes to assert that a regularity holds of certain entities in a natural language then the HPSGian expresses the assertion as a linguistic principle 'if A then C', where the consequent C is a description that expresses the regularity, and the antecedent A is a description that denotes the collection of entities of which the regularity is claimed to hold. If a HPSGian asserts that a regularity holds of certain entities in a natural language and expresses the assertion as an implicational description then the HPSGian is asserting that the description is true of each entity in the natural language, since, for each entity in the natural language, either the antecedent of the description is false of the entity, in which case the description is true of the entity, or the antecedent of the description is true of the entity, in which case the HPSGian asserts that the consequent of the description is true of the entity and hence that the description is true of the entity. I accordingly reformulate interim thesis 1 to include the simple intuition that a grammar is true of a natural language only if each description in the theory component of the grammar is true of each entity in the natural language.

INTERIM THESIS 2. *For each grammar* $\Gamma = (\Sigma, \theta)$, *for each natural language* I, *if* Γ *is true of* I

then I *is an interpretation of* Σ, *and*

I *models* θ *in* Σ.

Though the consequent of interim thesis 2 is certainly a stronger necessary condition for linguistic truth than the consequent of interim thesis 1, it is nonetheless *still* too weak. For example, the empty theory claims absolutely nothing of the entities in a natural language, and so a grammar comprising a signature and the empty theory cannot be true of a natural language other than vacuously. Yet interim thesis 2 may well judge such a grammar true of a natural language, since each interpretation models the empty theory. What the consequent of interim thesis 2 lacks is a relationship between a true grammar of a natural language and the entities *not* in the natural language. To illustrate this point, suppose that I is the English language, I' is I with every instance of the word "logic" and its derivatives ("logical", "logically", "logician", "logicism", "illogic", "illogical", "choplogic", etc.) systematically expunged, and $\Gamma = (\Sigma, \theta)$ is a true grammar of I. Then, by interim thesis 2, I is an interpretation of Σ and models θ in Σ. Thus, I' also is an interpretation of Σ and models θ in Σ. Thus, by interim thesis 2, Γ is true also of I'. However, intuition says that Γ must be false of I', and the basis of that intuition is clear: θ is true of some instances of the word "logic", yet no instances of the word "logic" are entities in I'. Γ is intuitively false of I' because θ is true of some linguistic token not in I'. A first reformulation of interim thesis 2 might therefore include the intuition that a grammar is true of a natural language only if the theory component of the grammar is false of each entity not in the natural language.

INTERIM THESIS 3. *For each grammar* $\Gamma = (\Sigma, \theta)$, *for each natural language* I,

if Γ *is true of* I

then I *is an interpretation of* Σ,

I *models* θ *in* Σ, *and*

for each interpretation I' *of* Σ, *for each* $\upsilon' \in I'$,

if $\upsilon' \in \Theta_{I'}(\theta)$ *then* $\upsilon' \in I$.

However, the consequent of interim thesis 3 is so much stronger than the consequent of its predecessor that the new thesis is false. Suppose that $\Gamma = (\Sigma, \theta)$ is a SRL expression of the (Pollard and Sag, 1994) HPSG grammar of English, entity υ is the quasi-English sentence "Kim walks

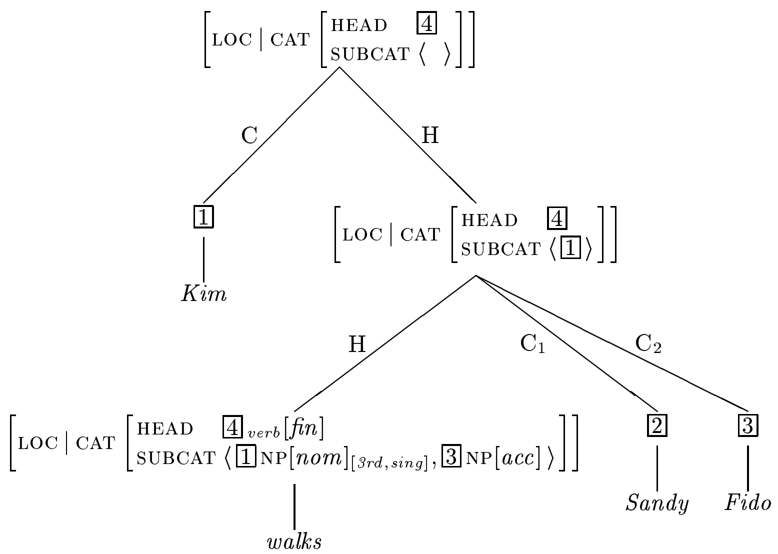


FIGURE 3. "Kim walks Sandy Fido"

Sandy Fido" with the structure partially depicted in figure 3,[9] and entity υ' is the head daughter of υ, the quasi-English verb phrase "walks Sandy Fido". Each principle in the (Pollard and Sag, 1994) grammar of English is true of υ, but the subcategorization principle is false of υ'. Thus, under interim thesis 3,

Γ is true of English

$\Longrightarrow$ English contains υ and does not contain υ'

$\Longrightarrow$ English contains υ' and does not contain υ'

$\Longrightarrow$ contradiction.

Now, Γ may indeed be false of English, but this alleged proof of its falseness is absurd. It therefore follows that a better reformulation of interim thesis 2 might include the intuition that a grammar is true of a natural language only if the theory component of the grammar is false of *some constituent of* each entity not in the natural language. For each signature Σ, for each interpretation I of Σ, C_I is the total function from U to $Pow(U)$ such that

[9]I use the abbreviatory conventions of (Pollard and Sag, 1994, pages 27, 28, 32 and 33) to depict the structure.

$$\text{for each } \upsilon \in I,\ C_I(\upsilon) = \left\{ \upsilon' \in I \;\middle|\; \begin{array}{l} \text{for some } \tau \in \mathsf{T}_\Sigma, \\ \quad T_I(\tau)(\upsilon) \text{ is defined, and} \\ \quad T_I(\tau)(\upsilon) = \upsilon' \end{array} \right\}.$$

I call each member of $C_I(\upsilon)$ a **constituent** of υ in I. Notice that this notion of constituency is more general than the traditional linguistic notion in that a first entity can be a constituent of a second regardless of whether either is a sign.

INTERIM THESIS 4. *For each grammar* $\Gamma = (\Sigma, \theta)$, *for each natural language* I,

if Γ *is true of* I

then I *is an interpretation of* Σ,

I *models* θ *in* Σ, *and*

for each interpretation I' *of* Σ, *for each* $\upsilon' \in I'$,

if $C_{I'}(\upsilon') \subseteq \Theta_{I'}(\theta)$ *then* $\upsilon' \in I$.

The consequent of interim thesis 4 is sufficiently weaker than the consequent of interim thesis 3 that the new thesis does not suffer from the absurdity of the old: interim thesis 4 does not falsify the (Pollard and Sag, 1994) grammar of English on the basis of "Kim walks Sandy Fido". However, the consequent of interim thesis 4 overlooks the possibility that the constituents of an object can be differently configured in different interpretations, that is, it overlooks the possibility that entity υ has exactly the same constituents in interpretations $I_1 = (U_1, S_1, F_1)$ and $I_2 = (U_2, S_2, F_2)$ of signature Σ yet for some $\tau \in \mathsf{T}_\Sigma$,

$T_{I_1}(\tau)(\upsilon)$ is defined but $T_{I_2}(\tau)(\upsilon)$ is undefined, or

$T_{I_1}(\tau)(\upsilon) \neq T_{I_2}(\tau)(\upsilon)$, or

$S_1(T_{I_1}(\tau)(\upsilon)) \neq S_2(T_{I_2}(\tau)(\upsilon))$.

Consequently, it can fail to judge a patently false grammar of a natural language false. For example, the English sentence "Adam thought Bill had crashed his car" is ambiguous, and so has several kinds of instance. In one kind of instance, Adam thought Bill had crashed Adam's car. In another, Adam thought Bill had crashed Bill's car. Suppose that I is the English language and $\Gamma = (\Sigma, \theta)$ is a true grammar of I. Under interim thesis 4,

I is an interpretation,

θ is true of each entity in I, and

θ is false of some constituent of each entity not in I.

Suppose further that I' is I with the constituents of each instance of "Adam thought Bill had crashed his car" in which Adam thought Bill had crashed Adam's car reconfigured into an instance of "Adam thought Bill had crashed his car" in which Adam thought Bill had crashed Bill's car. Γ is intuitively false of I', since I' has no instances of the sentence "Adam thought Bill had crashed his car" in which Adam thought Bill had crashed Adam's car. Yet interim thesis 4 fails to judge Γ false of I', since

I' is an interpretation,

θ is true of each entity in I', and

θ is false of some constituent of each entity not in I'.

An even better reformulation of interim thesis 2 than interim thesis 4 might therefore include the intuition that a grammar is true of a natural language only if the theory component of the grammar is false of some constituent of each entity *with a constituent configuration* not in the natural language. For each signature $\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$, for each interpretation $I_1 = (U_1, S_1, F_1)$ of Σ, for each interpretation $I_2 = (U_2, S_2, F_2)$ of Σ, for each υ,

I_1 and I_2 **share** υ in Σ

iff $\upsilon \in U_1$,

$\upsilon \in U_2$,

$C_{I_1}(\upsilon) = C_{I_2}(\upsilon)$,

for each $\upsilon' \in C_{I_1}(\upsilon)$, $S_1(\upsilon') = S_2(\upsilon')$, and

for each $\upsilon' \in C_{I_1}(\upsilon)$, for each $\varphi \in \mathsf{F}$,

$F_1(\varphi)(\upsilon')$ is defined iff $F_2(\varphi)(\upsilon')$ is defined, and

if $F_1(\varphi)(\upsilon')$ is defined then $F_1(\varphi)(\upsilon') = F_2(\varphi)(\upsilon')$.

INTERIM THESIS 5. *For each grammar $\Gamma = (\Sigma, \theta)$, for each natural language I,*

if Γ is true of I

then I is an interpretation of Σ,

I models θ in Σ, and

for each interpretation I' of Σ, for each $\upsilon' \in I'$,

if $C_{I'}(\upsilon') \subseteq \Theta_{I'}(\theta)$ then I and I' share υ' in Σ.

Interim thesis 5 seems all well and good, yet I still call it an *interim* thesis, and do so because it and its two predecessors have two simple shortcomings that easily escape notice. Firstly, if a grammar is true of a natural language under interim theses 3, 4 or 5 then the natural language must be either empty or a proper class. This shortcoming is an annoying but perhaps tolerable technicality. Unfortunately, the second shortcoming cannot be so lightly tolerated. If a grammar is true of a natural language under interim theses 3, 4 or 5 and the theory component of the grammar is true of each constituent of an entity in some interpretation then the entity must also be in the natural language. However, I later show in corollary 1 that if a nontrivial interpretation models a theory then a nontrivial interpretation consisting entirely of mathematical – hence nonlinguistic – entities also models the theory. Thus, if a grammar is true of a natural language under interim theses 3, 4 or 5 then the natural language must either be empty or contain nonlinguistic entities. An obvious circumvention of the second – though not perhaps the first – shortcoming might therefore seem to be a reformulation of interim thesis 2 that includes the intuition that a grammar is true of a natural language only if the theory component of the grammar is false of some constituent of each *linguistic* entity with a constituent configuration not in the natural language.

INTERIM THESIS 6. *For each grammar* $\Gamma = (\Sigma, \theta)$, *for each natural language* I,

if Γ *is true of* I
then I *is an interpretation of* Σ,
  I *models* θ *in* Σ, *and*
  for each interpretation I' *of* Σ, *for each linguistic* $\upsilon' \in I'$,
    if $C_{I'}(\upsilon') \subseteq \Theta_{I'}(\theta)$ *then* I *and* I' *share* υ' *in* Σ.

Unfortunately, this apparent circumvention is fundamentally and irretrievably flawed. At present, there can be no formal definition of what it is for an entity to be linguistic. Therefore, the consequent of interim thesis 6 is a necessary condition for linguistic truth that currently cannot be well-defined. I propose the following somewhat unusual escape from this predicament. I first present a formal definition of possibly distinct entities in possibly distinct interpretations being 'identical twins' in that the constituents of the identical twins within their respective interpretations are isomorphically configured. I then reformulate interim thesis 2 to include the intuition that a grammar is true of a natural language only if the theory component of the grammar is false of some constituent of each entity *without an identical twin* in the natural language. For each signature $\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$, for each interpretation $I_1 = (U_1, S_1, F_1)$ of Σ, for each interpretation $I_2 = (U_2, S_2, F_2)$ of Σ, for each υ_1, for each υ_2,

(υ_1, I_1) and (υ_2, I_2) are **congruent** in Σ
iff $\upsilon_1 \in U_1$,
$\upsilon_2 \in U_2$, and
for some bijection κ from $C_{I_1}(\upsilon_1)$ to $C_{I_2}(\upsilon_2)$,
$\kappa(\upsilon_1) = \upsilon_2$,
for each $\upsilon' \in C_{I_1}(\upsilon_1)$, $S_1(\upsilon') = S_2(\kappa(\upsilon'))$, and
for each $\upsilon' \in C_{I_1}(\upsilon_1)$, for each $\varphi \in \mathsf{F}$,
$F_1(\varphi)(\upsilon')$ is defined iff $F_2(\varphi)(\kappa(\upsilon'))$ is defined, and
if $F_1(\varphi)(\upsilon')$ is defined then $\kappa(F_1(\varphi)(\upsilon')) = F_2(\varphi)(\kappa(\upsilon'))$.

THESIS 1. *For each grammar* $\Gamma = (\Sigma, \theta)$*, for each natural language* I*,*
if Γ *is true of* I
then I *is an interpretation of* Σ*,*
I *models* θ *in* Σ*, and*
for each interpretation I' *of* Σ*, for each* $\upsilon' \in I'$*,*
if $C_{I'}(\upsilon') \subseteq \Theta_{I'}(\theta)$
then for some $\upsilon \in I$*,* (υ, I) *and* (υ', I') *are congruent in* Σ*.*

Thesis 1 is my preferred thesis. It formally expresses exactly three intuitions concerning the content of a grammar. A grammar (Σ, θ) is true of a natural language only if

the natural language can be construed as a system of linguistic tokens meeting the conditions imposed on them by Σ,

each description in θ is true of each entity in the natural language, and

some description in θ is false of some constituent of each entity for which no entity in the natural language has isomorphically configured constituents.

The extent to which a HPSGian shares these intuitions is exactly the extent to which thesis 1 captures the content of their grammars.

The form of thesis 1 is rather ugly and cumbersome. A better form turns upon an elegant equivalent of the congruence relation. For each signature Σ, for each interpretation I_1 of Σ, for each interpretation I_2 of Σ, for each $\upsilon_1 \in I_1$, for each $\upsilon_2 \in I_2$,

(υ_1, I_1) and (υ_2, I_2) are **indiscernible** in Σ

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

$\upsilon_1 \in D_{I_1}(\delta)$ iff $\upsilon_2 \in D_{I_2}(\delta)$.

PROPOSITION 1. *For each signature Σ, for each interpretation I_1 of Σ, for each interpretation I_2 of Σ, for each $\upsilon_1 \in I_1$, for each $\upsilon_2 \in I_2$,*

(υ_1, I_1) and (υ_2, I_2) are congruent in Σ

iff (υ_1, I_1) and (υ_2, I_2) are indiscernible in Σ.

For each signature Σ, for each interpretation I_1 of Σ, for each interpretation I_2 of Σ,

I_1 **simulates** I_2 in Σ

iff for each $\upsilon_2 \in I_2$, for some $\upsilon_1 \in I_1$,

(υ_1, I_1) and (υ_2, I_2) are indiscernible in Σ.

For each signature Σ, for each $\theta \subseteq \mathsf{D}_\Sigma$, for each interpretation I of Σ,

I **exhaustively models** θ in Σ

iff I models θ in Σ, and

for each interpretation I' of Σ,

if I' models θ in Σ then I simulates I' in Σ.

PROPOSITION 2. *For each signature Σ, for each $\theta \subseteq \mathsf{D}_\Sigma$, for each interpretation I of Σ,*

I exhaustively models θ in Σ

iff I models θ in Σ, and

for each interpretation I' of Σ, for each $\upsilon' \in I'$,

if $C_{I'}(\upsilon') \subseteq \Theta_{I'}(\theta)$

then for some $\upsilon \in I$, (υ, I) and (υ', I') are congruent in Σ.

I can now form a thesis that is equivalent to thesis 1 yet replaces the complex notions of constituency and congruence with the simpler notion of an exhaustive model.

THESIS 2. *For each grammar $\Gamma = (\Sigma, \theta)$, for each natural language I,*

if Γ is true of I

then I is an interpretation of Σ and

I exhaustively models θ in Σ.

Two issues remain outstanding. The consequent of interim thesis 5 is too strong in two respects. If a grammar is true of a nonempty natural language under the interim thesis then

the natural language must be a proper class, and

the natural language must contain nonlinguistic entities.

Now, congruence is weaker than sharing. If interpretations I_1 and I_2 share entity υ in signature Σ then (υ, I_1) and (υ, I_2) are congruent in Σ. Therefore, the consequent of thesis 1, and hence the consequent of thesis 2, is weaker than the consequent of interim thesis 5. Indeed, it is sufficiently weak to avoid the second shortcoming of the consequent of the interim thesis. But is it sufficiently weak to avoid the first? And is it still sufficiently strong to be linguistically useful? I address the second issue when I characterise verification in a later paper, but I conclude the present paper by addressing the first. I prescribe for each theory the construction of an exhaustive model. From this immediately follows the theorem that each theory has an exhaustive model. Since some theories have nontrivial models it immediately follows that some theories have nontrivial exhaustive models. Thus, if a grammar is true of a nonempty natural language under thesis 1 or thesis 2 then the natural language need not be a proper class. The construction of the exhaustive model is illuminating but somewhat technical, and some readers might prefer to turn directly to the statement of theorem 1 on page 48.

Central to my construction of an exhaustive model of a theory is the notion of a morph.[10] I write X^* for the set of finite strings comprising members of set X. For each signature $\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$, a **morph** in Σ is a triple (β, η, λ) such that

β is a nonempty subset of F^*,

for each $\omega \in \mathsf{F}^*$, for each $\varphi \in \mathsf{F}$, if $\omega\varphi \in \beta$ then $\omega \in \beta$,

η is an equivalence relation over β,

for each $\omega_1 \in \mathsf{F}^*$, for each $\omega_2 \in \mathsf{F}^*$, for each $\varphi \in \mathsf{F}$,

if $\omega_1\varphi \in \beta$ and $(\omega_1, \omega_2) \in \eta$ then $(\omega_1\varphi, \omega_2\varphi) \in \eta$,

λ is a total function from β to S,

for each $\omega_1 \in \mathsf{F}^*$, for each $\omega_2 \in \mathsf{F}^*$, if $(\omega_1, \omega_2) \in \eta$ then $\lambda(\omega_1) = \lambda(\omega_2)$,

for each $\omega \in \mathsf{F}^*$, for each $\varphi \in \mathsf{F}$,

if $\omega\varphi \in \beta$ then $\lambda(\omega\varphi) \in \mathsf{A}(\lambda(\omega), \varphi)$, and

[10](King, 1989) called morphs *descriptors*.

for each $\varphi \in \mathsf{F}$, for each $\omega \in \mathsf{F}^*$,

if $\omega \in \beta$ and $\mathsf{A}(\lambda(\omega), \varphi) \neq \emptyset$ then $\omega\varphi \in \beta$.

I write M_Σ for the set of morphs in Σ. The definition of a morph is rather long and technical, so I turn to automata theory to form a more intuitive picture of a morph. (Nerode, 1958) gives an elegant and well-known abstract representation of an automaton. The **Nerode representation** of an automaton α with an input alphabet I is a pair (β, η) such that

$$\beta = \left\{ \omega \in \mathsf{I}^* \,\middle|\, \alpha \text{ transits from its initial node on input } \omega \right\}, \text{ and}$$

$$\eta = \left\{ (\omega_1, \omega_2) \in \mathsf{I}^* \times \mathsf{I}^* \,\middle|\, \begin{array}{l} \text{for some node } \nu \text{ in } \alpha, \\ \quad \alpha \text{ transits from its initial node to } \nu \\ \qquad \text{on input } \omega_1, \text{ and} \\ \quad \alpha \text{ transits from its initial node to } \nu \\ \qquad \text{on input } \omega_2 \end{array} \right\}.$$

(Moore, 1956) offers an augmented notion of an automaton. A **Moore automaton** is an automaton that outputs a symbol from an output alphabet at each of its nodes. (Moshier, 1988) extends Nerode representation to an abstract representation of a Moore automaton. The **Moshier representation** of a Moore automaton α with an input alphabet I and an output alphabet O is a triple (β, η, λ) such that

(β, η) is the Nerode representation of α, and

$$\lambda = \left\{ (\omega, o) \in \mathsf{I}^* \times \mathsf{O} \,\middle|\, \begin{array}{l} \text{for some node } \nu \text{ in } \alpha, \\ \quad \alpha \text{ transits from its initial node to } \nu \\ \qquad \text{on input } \omega, \text{ and} \\ \quad \alpha \text{ outputs } o \text{ at } \nu \end{array} \right\}.$$

It is usually stipulated that an automaton must have a finite set of nodes, a finite transition function and a finite input alphabet, and that a Moore automaton must have a finite output alphabet, but let us put these stipulations aside, and allow automata to have possibly infinite node sets, transition functions and alphabets. For each signature $\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$, for each Moore automaton α,

α is **totally well-typed** in Σ

iff α has input alphabet F,

α has output alphabet S,

for each node ν in α, for each $\varphi \in \mathsf{F}$,

α transits from ν on input φ

iff for some $\sigma \in \mathsf{S}$,

α outputs σ at ν and $\mathsf{A}(\sigma, \varphi) \neq \emptyset$, and

for each node ν in α, for each $\varphi \in \mathsf{F}$,
if α transits from ν on input φ
then for some node ν' in α, for some $\sigma \in \mathsf{S}$, for some $\sigma' \in \mathsf{S}$,
α transits from ν to ν' on input φ,
α outputs σ at ν,
α outputs σ' at ν', and
$\sigma' \in \mathsf{A}(\sigma, \varphi)$.

A morph is the Moshier representation of a totally well-typed Moore automaton.

For each signature $\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$, for each $\mu = (\beta, \eta, \lambda) \in \mathsf{M}_\Sigma$, for each $\omega \in \beta$,

$\beta/\omega = \{\omega' \in \mathsf{F}^* \mid \omega\omega' \in \beta\}$,
$\eta/\omega = \{(\omega_1, \omega_2) \in \mathsf{F}^* \times \mathsf{F}^* \mid (\omega\omega_1, \omega\omega_2) \in \eta\}$,
$\lambda/\omega = \{(\omega', \sigma) \in \mathsf{F}^* \times \mathsf{S} \mid (\omega\omega', \sigma) \in \lambda\}$, and
$\mu/\omega = (\beta/\omega, \eta/\omega, \lambda/\omega)$.

PROPOSITION 3. *For each signature* $\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$, *for each* $\mu = (\beta, \eta, \lambda) \in \mathsf{M}_\Sigma$, *for each* $\omega \in \beta$,

$\mu/\omega \in \mathsf{M}_\Sigma$.

I call μ/ω the ω **submorph** of μ in Σ. If μ is the Moshier representation of totally well-typed Moore automaton α then α transits from its initial node to some node ν on input ω and μ/ω is the Moshier representation of the totally well-typed Moore subautomaton derived from α with initial node ν.

For each signature $\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$, M_Σ is the total function from D_Σ to $Pow(\mathsf{M}_\Sigma)$, and A_Σ is the total function from $Pow(\mathsf{D}_\Sigma)$ to $Pow(\mathsf{M}_\Sigma)$ such that

for each $\omega_1 \in \mathsf{F}^*$, for each $\omega_2 \in \mathsf{F}^*$,
$M_\Sigma(:\omega_1 \approx :\omega_2) = \left\{(\beta, \eta, \lambda) \in \mathsf{M}_\Sigma \,\middle|\, (\omega_1, \omega_2) \in \eta \right\}$,
for each $\omega \in \mathsf{F}^*$, for each $\sigma \in \mathsf{S}$,
$M_\Sigma(:\omega \sim \sigma) = \left\{(\beta, \eta, \lambda) \in \mathsf{M}_\Sigma \,\middle|\, (\omega, \sigma) \in \lambda \right\}$,
for each $\delta \in \mathsf{D}_\Sigma$, $M_\Sigma(\neg\delta) = \mathsf{M}_\Sigma \setminus M_\Sigma(\delta)$,
for each $\delta_1 \in \mathsf{D}_\Sigma$, for each $\delta_2 \in \mathsf{D}_\Sigma$, $M_\Sigma([\delta_1 \wedge \delta_2]) = M_\Sigma(\delta_1) \cap M_\Sigma(\delta_2)$,
for each $\delta_1 \in \mathsf{D}_\Sigma$, for each $\delta_2 \in \mathsf{D}_\Sigma$, $M_\Sigma([\delta_1 \vee \delta_2]) = M_\Sigma(\delta_1) \cup M_\Sigma(\delta_2)$,

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

$$M_\Sigma([\delta_1 \to \delta_2]) = (\mathsf{M}_\Sigma \setminus M_\Sigma(\delta_1)) \cup M_\Sigma(\delta_2), \text{ and}$$

for each $\theta \subseteq \mathsf{D}_\Sigma$,

$$A_\Sigma(\theta) = \left\{ (\beta, \eta, \lambda) \in \mathsf{M}_\Sigma \,\middle|\, \begin{array}{c} \text{for each } \delta \in \theta, \text{ for each } \omega \in \beta, \\ (\beta, \eta, \lambda)/\omega \in M_\Sigma(\delta) \end{array} \right\}.$$

If $\mu \in M_\Sigma(\delta)$ then I say that δ **approximates** μ in Σ, and if $\mu \in A_\Sigma(\theta)$ then I say that θ **admits** μ in Σ. Admission is a recursive application of approximation: theory θ admits morph μ iff each description in θ approximates not only μ but also every submorph of μ.[11]

For each signature Σ, a **canonical entity** in Σ is a quadruple $(\beta, \eta, \lambda, \varepsilon)$ such that $(\beta, \eta, \lambda) \in \mathsf{M}_\Sigma$ and ε is an equivalence class under η. I write E_Σ for the set of canonical entities in Σ. Moshier representation extends to the abstract representation of Moore automata with distinguished nodes. The **Moshier representation** of a Moore automaton α with an input alphabet I, an output alphabet O, and a distinguished node ν is a quadruple $(\beta, \eta, \lambda, \varepsilon)$ such that

(β, η, λ) is the Moshier representation of α, and

$$\varepsilon = \left\{ \omega \in \mathsf{I}^* \,\middle|\, \alpha \text{ transits from its initial node to } \nu \text{ on input } \omega \right\}.$$

[11] There are obvious similarities between morphs and approximation on the one hand and (Carpenter, 1992) feature structures and satisfaction on the other. However, these similarities do not justify the (Pollard and Sag, 1994) vision of founding HPSG on a logic very similar to that of (Carpenter, 1992), since the behaviour of morphs and feature structures radically differ in crucial ways. For example, morph subsumption is extremely degenerate, in that morph $(\beta_1, \eta_1, \lambda_1)$ subsumes morph $(\beta_2, \eta_2, \lambda_2)$ iff $\beta_1 = \beta_2$, $\eta_1 \subseteq \eta_2$ and $\lambda_1 = \lambda_2$. Applying (Carpenter, 1992) terminology to morphs, if a first morph subsumes a second morph then the two morphs can differ only in that the second may have more shared structure than the first. It can even be cogently argued that morph subsumption should be fully degenerate, and that morph μ_1 should subsume morph μ_2 iff $\mu_1 = \mu_2$. In either case, morphs cannot play the role of models of partial information that (Carpenter, 1992) intends its feature structures to play. Moreover, the syntactic, semantic and logical apparatus (Carpenter, 1992) institutes for its feature structures does not apply to morphs. For example, a (Carpenter, 1992) description can have feature structure satisfiers yet no morph satisfiers. It is interesting to note, however, that if morph subsumption *is* fully degenerate then morphs become excellent models not of *partial* information but of *total* information, consistent information that cannot be consistently extended. On this view, morphs provide a valuable point of contact between entity-based logics such as SRL and information-based logics such as that of (Carpenter, 1992). They occupy an intermediary position between the two logic families, being neither entities nor models of partial information, yet related to both. Some papers, (Gerdemann and King, 1994) and (King, 1994) for example, have exploited this position in order to adapt computational techniques developed for information-based logics to entity-based logics more suited to HPSG.

A canonical entity is the Moshier representation of a totally well-typed Moore automaton with a distinguished node.

Using canonical entities, I can at last construct my exhaustive model. For each signature $\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$, for each grammar $\Gamma = (\Sigma, \theta)$,

$\mathcal{U}_\Gamma = \left\{ (\beta, \eta, \lambda, \varepsilon) \in \mathsf{E}_\Sigma \,\middle|\, (\beta, \eta, \lambda) \in A_\Sigma(\theta) \right\}$,

$\mathcal{S}_\Gamma = \left\{ ((\beta, \eta, \lambda, \varepsilon), \sigma) \in \mathcal{U}_\Gamma \times \mathsf{S} \,\middle|\, \text{for some } \omega \in \varepsilon,\ \lambda(\omega) = \sigma \right\}$,

$\mathcal{F}_\Gamma$ is the total function from F to $Pow(\mathcal{U}_\Gamma \times \mathcal{U}_\Gamma)$ such that for each $\varphi \in \mathsf{F}$, $\mathcal{F}_\Gamma(\varphi) = \left\{ ((\beta, \eta, \lambda, \varepsilon), (\beta, \eta, \lambda, \varepsilon')) \in \mathcal{U}_\Gamma \times \mathcal{U}_\Gamma \,\middle|\, \begin{array}{c} \text{for some } \omega \in \varepsilon, \\ \omega\varphi \in \varepsilon' \end{array} \right\}$, and

$\mathcal{I}_\Gamma = (\mathcal{U}_\Gamma, \mathcal{S}_\Gamma, \mathcal{F}_\Gamma)$.

PROPOSITION 4. *For each grammar* $\Gamma = (\Sigma, \theta)$,

$\mathcal{I}_\Gamma$ *is an interpretation of* Σ, *and*

$\mathcal{I}_\Gamma$ *exhaustively models* θ *in* Σ.

I call $\mathcal{I}_\Gamma$ the **canonical interpretation** in Γ. A clearer picture of a canonical interpretation is perhaps achieved if I adopt a modest terminological fiction, and speak of a canonical entity as a Moore automaton with a distinguished node rather than as the Moshier representation of a Moore automaton with a distinguished node. Under this fiction, if $\Gamma = (\Sigma, \theta)$ is a grammar then

an entity is in $\mathcal{I}_\Gamma$ iff the entity is a totally well-typed Moore automaton α with a distinguished node such that θ admits α,

Moore automaton α with distinguished node ν is in species σ iff α outputs σ at ν, and

feature φ maps Moore automaton α with distinguished node ν to Moore automaton α' with distinguished node ν' iff $\alpha = \alpha'$ and α transits from ν to ν' on input φ.

An immediate corollary of proposition 4 is:

COROLLARY 1. *For each grammar* $\Gamma = (\Sigma, \theta)$,

if for some interpretation I *of* Σ, I *is nontrivial and models* θ *in* Σ

then $\mathcal{I}_\Gamma$ *is nontrivial and models* θ *in* Γ.

On page 40, I used corollary 1 to show that if a grammar is true of a natural language under interim theses 3, 4 or 5 then the natural language must either be empty or contain nonlinguistic entities. Another immediate corollary of proposition 4 is:

THEOREM 1. *For each grammar* $\Gamma = (\Sigma, \theta)$,
for some interpretation I *of* Σ, I *exhaustively models* θ *in* Σ.

Those readers troubled that *every* theory has an exhaustive model should note that SRL does not forbid trivial interpretations. Any theory incoherent enough to preclude nontrivial models will have only trivial exhaustive models, but any theory coherent enough to allow nontrivial models will have only nontrivial exhaustive models. Coherent theories abound. For example, the empty theory has nontrivial models in almost all signatures. Thus, unlike under interim thesis 5, if a grammar is true of a natural language under thesis 1 or thesis 2 then the natural language need not necessarily be empty or a proper class. However, SRL cannot prevent incoherence. The burden of writing a coherent grammar falls fully and fairly on the HPSGian.

ACKNOWLEDGEMENTS

This paper has been far too long in preparation, and I have accrued a huge debt of gratitude to an extensive list of people in the process of researching and writing it. As partial expiation, the least I can do is thank those of my creditors I can remember, and apologise to those I cannot. So, I give heartfelt thanks to Steve Abney, Bjørn Aldag, Patrick Blackburn, Bob Carpenter, Dale Gerdemann, John Griffith, Thilo Götz, Yasunari Harada, Erhard Hinrichs, Stephan Kepser, Valia Kordoni, Peter Krause, Detmar Meurers, Uwe Mönnich, Drew Moshier, Tsuneko Nakazawa, Stanley Peters, Adam Przepiórkowski, Stephan Riezler, Sabine Reinhart, Ivan Sag, Manfred Sailer, Kiril Simov, Jürgen Wedekind and the students who attended my various classes on HPSG formalism at Stanford University, Tübingen University and the 1996 European Summer School in Logic, Language and Information. I particularly wish to thank Tilman Höhle, Carl Pollard and Frank Richter for sometimes heated but always illuminating discussion of the research this paper reports. Of course, all remaining mistakes are mine alone. At various stages in researching and writing this paper, I received financial assistance from the Systems Development Foundation, the Deutsche Forschungsgemeinschaft, and the Volkwagen-Stiftung, and I wish to express my gratitude to these institutions for their support. Finally, I thank Jennifer, *Emma*, Heather, *Blump*, *Tiny* and Erica, without whom the writing of this paper would have been entirely possible but utterly pointless.

References

Aït-Kaci, H. and Nasr, R. (1986). LOGIN: A logical programming language with built-in inheritance. *Journal of Logic Programming*, 3:187–215.

Carpenter, B. (1990). Typed feature structures: Inheritance, (in)equations, and extensionality. In *Proceedings of the First International Workshop on Inheritance in Natural Language Processing*, pages 9–13, Tilburg, The Netherlands.

Carpenter, B. (1992). *The Logic of Typed Feature Structures*. Number 32 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, England.

Carpenter, B. and Pollard, C. (1991). Inclusion, disjointness and choice: The logic of linguistic classification. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 9–16, Morristown, New Jersey, USA.

Carpenter, B., Pollard, C., and Franz, A. (1991). The specification and implementation of constraint-based unification grammars. In *Proceedings of the Second International Workshop on Parsing Technology*, pages 143–153, Cancun, Mexico.

Chomsky, N. (1957). *Syntactic Structures*. Number 4 in Janua Linguarum, Series Minor. Mouton Publishers, The Hague, The Netherlands.

Chomsky, N. (1986). *Knowledge of Language: Its Nature, Origin and Use*. Ruth Nanda Anshen's 'Convergence' Series. Praeger Publishers, New York, New York, USA.

Gazdar, G., Pullum, G. K., Carpenter, R., Klein, E., Hukari, T. E., and Levine, R. D. (1988). Category structures. *Computational Linguistics*, 14(1):1–19.

Gerdemann, D. and King, P. J. (1994). The correct and efficient implementation of appropriateness specifications for typed feature structures. In *Proceedings of COLING 94*, pages 956–960, Kyoto, Japan.

Höhfeld, M. and Smolka, G. (1988). Definite relations over constraint languages. LILOG technical report 53, IBM Deutschland GmbH, Stuttgart, Germany.

Höhle, T. N. (1999). An architecture for phonology. In Borsley, R. D. and Przepiórkowski, A., editors, *Slavic in HPSG*. CSLI, Stanford, California, USA.

Johnson, M. (1988). *Attribute-Value Logic and the Theory of Grammar*. Number 16 in CSLI Lecture Notes. CSLI, Stanford, California, USA.

Johnson, M. (1991). Features and formulae. *Computational Linguistics*, 17(1):131–152.

Kasper, R. T. and Rounds, W. C. (1986). A logical semantics for feature structures. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, pages 257–266, New York, New York, USA.

Kepser, S. (1994). A satisfiability algorithm for a typed feature logic. Sonderforschungsbereich 340 technical report 60, Eberhard-Karls-Universität, Tübingen, Germany.

King, P. J. (1989). *A Logical Formalism for Head-Driven Phrase Structure Grammar*. PhD thesis, Manchester University, Manchester, England.

King, P. J. (1994). Typed feature structures as descriptions. In *Proceedings of COLING 94*, pages 1250–1254, Kyoto, Japan.

King, P. J., Simov, K. I., and Aldag, B. (1999). The complexity of modellability in finite and computable signatures of a constraint logic for head-driven phrase structure grammar. *The Journal of Logic, Language and Information*, 8(1):83–110.

Meurers, W. D. and Minnen, G. (1997). A computational treatment of lexical rules in HPSG as covariation in lexical entries. *Computational Linguistics*, 23(4):543–568.

Moore, E. F. (1956). Gedanken experiments on sequential machines. In *Automata Studies*, pages 129–153. Princeton University Press, Princeton, New Jersey, USA.

Moshier, M. A. (1988). *Extensions to Unification Grammar for the Description of Programming Languages*. PhD thesis, University of Michigan, Ann Arbor, Michigan, USA.

Moshier, M. D. and Rounds, W. C. (1987). A logic for partially specified data structures. In *Proceedings of the 14th ACM Symposium on Principles of Programming Languages*, pages 156–167, München, Germany.

Nerode, A. (1958). Linear automaton transformations. *Proceedings of the American Mathematical Society*, 9:541–544.

Pollard, C. J. (1989). Sorts in unification-based grammar and what they mean. Unpublished manuscript.

Pollard, C. J. (1999). Strong generative capacity in HPSG. In Webelhuth, G., Koenig, J.-P., and Kathol, A., editors, *Lexical and Constructional Aspects of Linguistic Explanation*, pages 281–297. CSLI, Stanford, California, USA.

Pollard, C. J. and Carpenter, B. (1990). Extensionality in feature structures and feature logic. Paper presented at the Workshop on Unification and Generation, Bad Teinach, Germany.

Pollard, C. J. and Moshier, M. D. (1994). Unifying partial descriptions of sets. In Hanson, P. P., editor, *Information, Language and Cognition*, pages 285–322. Oxford University Press, Oxford, England.

Pollard, C. J. and Sag, I. A. (1987). *Information-Based Syntax and Semantics*. Number 13 in CSLI Lecture Notes. CSLI, Stanford, California, USA.

Pollard, C. J. and Sag, I. A. (1994). *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago, Illinois, USA.

Richter, F. (1999). RSRL for HPSG. In this volume.

Richter, F. and Sailer, M. (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, Germany.

Richter, F., Sailer, M., and Penn, G. (1999). A formal interpretation of relations and quantification in HPSG. In this volume.

Saussure, F. (1916). *Cours de Linguistique Générale*. Editions Payot, Paris, France.

Shieber, S. M. (1986). *An Introduction to Unification-based Approaches to Grammar*. Number 4 in CSLI Lecture Notes. CSLI, Stanford, California, USA.

Smolka, G. (1988). A feature logic with subsorts. LILOG technical report 33, IBM Deutschland GmbH, Stuttgart, Germany.

The CLaRK Programme, Seminar für Sprachwissenschaft, Eberhard-Karls-Universität, Tübingen, Germany