Introduction[1]

GEORGIA M. GREEN *University of Illinois*
ROBERT D. LEVINE *Ohio State University*

1 Overview

Head-driven phrase structure grammar (HPSG) has evolved as a synthesis of ideas from a number of theoretical sources, including generalized phrase structure grammar (GPSG), categorial grammar, and formal theories of data structure representation (e.g., the PATR-II formalism and Kasper-Rounds logics of computation). In the course of more than a decade of development these theoretical resources have been applied to a monostratal theory of linguistic structure which is capable of providing a formally explicit grammar for any given natural language. HPSG uses a fundamental theoretical strategy made familiar by GPSG: the enumeration of a class of objects, corresponding to expressions of some natural language, and a set of constraints whose interaction enforces the appropriate covariation of formal properties reflecting the dependencies that any grammar of that language must capture.

A head-driven phrase structure grammar of some language defines the set of **signs** (form/meaning correspondences) which that language comprises. The formal entities that model signs in HPSG are complex objects called **feature structures**, whose form is limited by a set of constraints – some universal and some language-parochial. The interaction of these constraints defines the grammatical structure of each such sign and the morphosyntactic dependencies which hold between its subcomponents. Given a specific set of such constraints, and a lexicon providing at least one feature structure description for each word in the language, an infinite number of signs is recursively characterized.

[1] Green's work was supported in part by the Beckman Institute for Advanced Science and Technology at the University of Illinois at Urbana-Champaign.

It is useful to distinguish three phases in the evolution of what may be thought of as a "classical" theory of HPSG: the version presented in Pollard and Sag (1987), informally referred to as HPSG-I; chapters 1–8 of Pollard and Sag (1994) defining HPSG-II, and chapter 9 of Pollard and Sag (1994) containing enough revisions of the theory offered in the preceding part of the volume to constitute a separate version of the theory, HPSG-III. As HPSG has become adopted by an increasing number of investigators, there have been numerous innovations and emendations grafted onto this basic theory, many of which are proposed, illustrated or defended in the contributions in this volume. At present, the theory is in an intense and fertile period of development which precludes the possibility of a straightforward unitary treatment, but the following chapters afford a useful point of departure for those who wish to acquaint themselves with current thinking in this framework. To assist readers in this pursuit, we offer in section 2 below a basic introduction to some of the leading concepts of the classical theory, and in section 3 outline the enrichments, extensions, and revisions to this theory contributed by each of the papers in this collection.

2 Fundamentals of HPSG

2.1 *Feature structures, signs, and types*

2.1.1 *Feature structures and feature structure descriptions*

All linguistic objects (including both expression types, and the abstract objects that are invoked to describe them) are modeled in HPSG as **feature structures**.[2] A feature structure is a complete specification of all the properties of the object it models. Feature specifications consist of a value of the appropriate kind, or **type**, for each required attribute or **feature**. In other words, all attributes of the object being modeled must be specified.

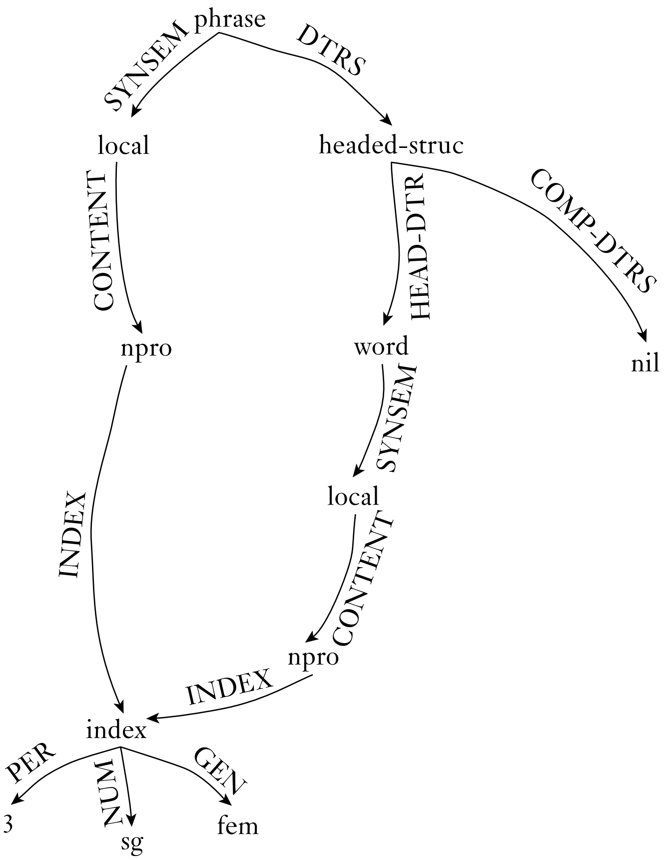
Feature structures themselves are represented as directed graphs, not necessarily acyclic, subject to certain formal restrictions.[3]

A schematic example of a feature structure is given in (1):

[2] For discussion see Pollard and Sag (1994: 8, 17–18), and for background, Shieber (1986), Pollard and Moshier (1990), Carpenter (1992).

[3] For example, they must be totally well-typed, which means that they are complete models rather than partial models (i.e., constraints or descriptions) of the objects they represent. Feature structures must also be sort-resolved, which is to say that all values must be the maximal (most specific) ones possible; i.e., every node q in the feature structure must be labeled by a sort with no subsorts, and every path must terminate in an atomic sort, one with no attribute declarations.

(1)



This (radically oversimplified) graph, reflecting a typical HPSG-II representation, is, very roughly speaking, a (partial) representation of the non-branching, headed phrase structure (2):

(2)

$$\begin{array}{c} XP_i \\ | \\ X_i \end{array}$$

The feature structure in (1) reflects the following information: the sign in question is of subtype *phrase*, whose head daughter is of subtype *word*, i.e., a lexical sign, specified for a feature INDEX as part of its semantic attributes, indicated by the feature CONTENT; furthermore, as the convergence of the arrows indicates, the INDEX of the head daughter is explicitly identified as being the same thing as the index of the mother phrase itself.

The graph representation of feature structures is awkward to display and tedious to interpret, so, as a convenience, feature structure descriptions in the form of attribute value matrices (AVMs) are commonly used instead. Attribute or feature names are typically written in upper case in AVMs, representing feature structure descriptions, and values are written to the right of the feature name, in lower case as in (3).

(3)
$$\begin{bmatrix} \text{PER} & \mathit{3} \\ \text{NUM} & \mathit{sg} \\ \text{GEN} & \mathit{fem} \end{bmatrix}$$

Types of feature structures inherit all of the attributes and type restrictions on their values from all of their supertypes.[4,5]

Feature structures are the entities constrained by the grammar. It is crucially important to distinguish between feature structures (fully specified objects) and **feature structure descriptions**, objects that (partially) describe feature structures. Feature structure descriptions characterize classes of objects. For example, the NP *she* could be represented by a fully specified feature structure (representable as a directed graph), but "NP" is (an abbreviation for) a feature structure description, and, under the restrictions described in note 3, could not be so represented. Put another way, a partial description is a constraint on members of a class of feature structures, while a total description is a constraint which limits the class to a single member. For the most part, grammar specification deals with generalizations over classes of words and phrases, and therefore with (partial) feature structure descriptions.

2.1.2 Signs and their attributes

As already noted, the primary object of linguistic analysis in HPSG is the sign, which models the association of form and meaning. Signs belong to one of two types: *word* and *phrase*. An act of uttering a linguistic expression corresponding to a particular sign is an act of producing a noise that corresponds to the phonological properties of that sign, with the intent that the product of that act be understood as intended to have syntactic, semantic, and contextual properties corresponding to the respective attributes of that sign.

The sign itself is an abstract structured object with phonological, syntactico-semantic, and contextual attributes, expressing different kinds of properties of the sign.

[4] In previous formulations of HPSG, this inheritance is strictly monotonic; adding information must never entail the revision of specifications. In recent work by Sag (1997) this requirement is relaxed.

[5] The set of feature-structure types in a grammar is a partial subsumption ordering, i.e., a transitive, reflexive, and antisymmetric relation on the subsumption relation. Thus, the division of the broad class of signs into words and phrases noted above represents the fact that the type *sign* subsumes both *phrase* and *word*. In fact, since the specifications for *phrase* and *word* are mutually exclusive (phrases have attributes which specify their immediate constituents, and words don't), the types *phrase* and *word* PARTITION the type *sign*.

The feature system employed in HPSG echoes GPSG feature theory in a number of respects, but there are some significant differences. One particularly evident difference between the two frameworks is that HPSG's feature geometry is considerably more ramified. In HPSG, all signs are assumed to have PHON and SYNSEM attributes, recording their phonological and syntactico-semantic structures, respectively.[6] The value of the SYNSEM attribute is a feature structure which represents the constellation of properties that can be subcategorized for. It has a LOCAL attribute, whose value has CATEGORY, CONTENT, and CONTEXT attributes, and represents what is shared by filler and gap in so-called extraction constructions. It also has a NONLOCAL attribute, whose values constrain all types of unbounded dependency constructions (UDCs).

The CATEGORY attribute takes as its value a category, whose attribute HEAD has as its value a part of speech and whose valence attributes SUBJ, COMPS, and SPR each have a list of synsems as their value-type. An ARG-ST feature is a property of all words. Its value is a list of synsems denoting the sign's arguments, ordered by obliqueness, and it contains the obliqueness record that is invoked in constraining binding relations (cf. Pollard and Sag 1994: ch. 6). The valence attributes take over the saturation-tracking function of the HPSG-II SUBCAT feature.

The value of the CONTENT feature is (depending on which part of speech the HEAD value is) a nominal object, a quantifier or a parameterized-state-of affairs (a *psoa*). A psoa is (roughly speaking) a representation of a (possibly open) proposition. Psoas form an elaborate type subhierarchy, with kinds of relations naming subtypes and determining what argument-denoting attributes they have, as illustrated in (4).[7]

(4)
$$\begin{bmatrix} \textit{nominate} & \\ \text{NOMINATOR} & \textit{index} \\ \text{NOMINEE} & \textit{index} \end{bmatrix} \begin{bmatrix} \textit{persuade} & \\ \text{PERSUADER} & \textit{index} \\ \text{PERSUADEE} & \textit{index} \\ \text{STATE-OF-AFFAIRS} & \textit{psoa} \end{bmatrix}$$

A nominal object, by contrast, corresponds to the logical representation of a common noun (although it is useful, in certain cases, to take common nouns to have semantic values representable as psoas, particularly when the noun denotes an event rather than an individual, as discussed in Michael

6 PHON values are usually represented in standard orthography, solely for the sake of convenience and readability.

7 The representation of propositional content as psoas does not reflect an essential property of HPSG. It would make no difference if some other kind of coherent representation of a semantic analysis was substituted, as long as it provided a way of indicating what properties can be predicated of which arguments, how arguments are linked to individuals in a model, and how the meaning of each constituent is a function of the meaning of its parts. In other words, the exact form of the representation is not crucial as long as it provides a compositional semantics.

Johnston's paper in this volume). It is expressed in a feature structure representation containing an INDEX attribute, with values of type *index*, and a RESTRICTION attribute, whose value is a set of psoas. Indexes in turn have attributes for PERSON, NUMBER, and GENDER. For perspicuity, in abbreviated AVMs, index values are often represented as subscripts on category designations: NP_i, for example, or NP_{there}. A psoa-valued CONTENT specification is similarly abbreviated following a colon after a category designation; VP:[i] represents a VP with the content [i].

Finally, the CONTEXT attribute records indexical information (in the values of the SPEAKER, ADDRESSEE, and LOCATION features) and presuppositions in the psoa-set value of the BACKGROUND attribute. Linguistically relevant information that is generally considered pragmatic is supposed to be represented in the value of the CONTEXT attribute. For some discussion, see Green (1995).

2.1.3 Types

The requirements of well-typedness and sort-resolution (see note 3) entail that grammars must be complete and explicit about what kinds of features are required to properly characterize an object of any given type, and also what kind of objects (i.e., what types) are appropriate values for any given feature. The CONTENT attribute of a sign, for example, can have, among other possible types, an object of type *nonpronominal* (or *npro*) as its value, and *npro*-type objects require, inter alia, a specification for an INDEX value. Thus, a sort is defined by a declaration of the attributes (features) it has, and the value-types of those features. Feature declarations are represented as labeled attribute-value matrices, AVMs, as illustrated in (5), where F_i are feature names and $sort_i$ are sort names.

(5) $$\begin{bmatrix} \mathit{sort_0} & \\ F_1 & \mathit{sort_1} \\ F_2 & \mathit{sort_2} \\ \ldots & \\ F_n & \mathit{sort_n} \end{bmatrix}$$

Sort declarations specify what attributes an instance of the sort has, and what kinds of things the values of those attributes can be, and sometimes what particular value an attribute must have (either absolutely, or relative to the value of some other attribute). For two sorts, *a* and *b*, *a* is a subsort of *b* iff it is dominated by *b* in a hierarchical classification of sorts generally referred to as the sort hierarchy; sorts which label terminal nodes in the sort hierarchy are termed "maximal sorts" because they are maximally informative or specific.

Constraints on feature structures are expressed in terms of feature-structure descriptions, and can therefore take full advantage of underspecification and subsumption relations. What is implicit in sort definitions (including lexical specifications) or in universal or language-specific constraints does not have to be expressed in the representations of linguistic objects. For example, since the Head Feature Principle requires that the HEAD value of the head daughter of a phrase be the same as the HEAD value of the phrase itself, the details of this value only need to be indicated once in each representation of a phrase. The notion of the values of two attributes being the very same object is modeled in feature structures as the sharing of structure, as illustrated above in (1). In referring to token-identity, and not just type-matching, STRUCTURE-SHARING is a crucial property of HPSG which does not have a direct counterpart in other syntactic theories. It amounts to the claim that the value of some instance of an attribute is token-identical to the value of some other instance of an attribute, i.e., it is the SAME THING – not a different thing which happens to have all the same properties. As indicated in (1) above, structure-sharing is represented in feature structures as a convergence of arrows (sometimes referred to as re-entrancy), whereas in AVMs this kind of token-identity is shown via recurrence of TAGS – boxed integers like $\boxed{3}$.

Thus, the following three AVMs are equivalent descriptions of the feature structure in (1):[8]

(6) a.
$$\left[\begin{array}{ll} \text{SYNSEM}\,|\,\text{CONTENT}\,|\,\text{INDEX}\ \boxed{1}\left[\begin{array}{ll}\text{PER} & \mathit{3}\\ \text{NUM} & \mathit{sg}\\ \text{GEN} & \mathit{fem}\end{array}\right] \\ \text{DTRS}\ \left[\begin{array}{l}\text{HEAD-DTR}\,|\,\text{SYNSEM}\,|\,\text{CONT}\,|\,\text{INDEX}\ \boxed{1}\\ \text{COMPS-DTRS}\ \mathit{nil}\end{array}\right]\end{array}\right]$$

b.
$$\left[\begin{array}{l} \text{SYNSEM}\,|\,\text{CONTENT}\,|\,\text{INDEX}\ \boxed{1} \\ \text{DTRS}\ \left[\begin{array}{l}\text{HEAD-DTR}\,|\,\text{SYNSEM}\,|\,\text{CONT}\,|\,\text{INDEX}\ \boxed{1}\left[\begin{array}{ll}\text{PER} & \mathit{3}\\ \text{NUM} & \mathit{sg}\\ \text{GEN} & \mathit{fem}\end{array}\right]\\ \text{COMPS-DTRS}\ \mathit{nil}\end{array}\right]\end{array}\right]$$

c.
$$\left[\begin{array}{l} \text{SYNSEM}\,|\,\text{CONTENT}\,|\,\text{INDEX}\ \boxed{1}\,\text{NUM}\ \mathit{sg} \\ \text{DTRS}\ \left[\begin{array}{l}\text{HEAD-DTR}\,|\,\text{SYNSEM}\,|\,\text{CONT}\,|\,\text{INDEX}\ \boxed{1}\left[\begin{array}{ll}\text{PER} & \mathit{3}\\ \text{GEN} & \mathit{fem}\end{array}\right]\\ \text{COMPS-DTRS}\ \mathit{nil}\end{array}\right]\end{array}\right]$$

All three descriptions convey the same information, since there is a only one way to satisfy the token-identities in the three descriptions.[9]

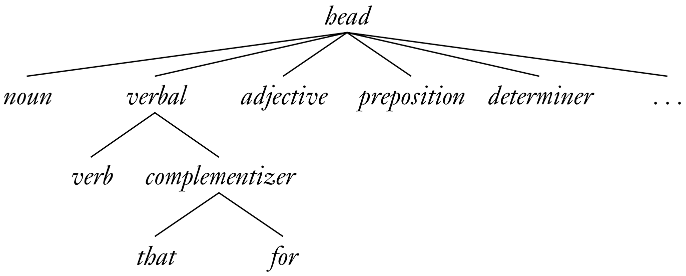
As noted above, types, or sorts, are organized hierarchically in the logic of HPSG. For each local subtree in the type hierarchy, the sorts which label the daughters partition the sort which labels the mother; that is, they are

[8] In the following AVMs we employ the conventional representation of feature-name pathways in which [A [B [C *x*]]] is represented as A | B | C *x*. For reasons of perspicuity, sometimes values are labeled with the name (in *italics*) of the sort that structures their content, but such information is usually omitted wherever possible.

[9] Note that in certain recent work DTRS is not employed; rather, HEAD-DTR and NONHEAD-DTRS are "top-level" attributes of phrasal signs.

necessarily disjoint subsorts which exhaust the sort of the mother. For example, subsorts of *head* can be any of a number of part-of-speech types, of which some are both further partitioned, as illustrated in (7), following Sag (1997).

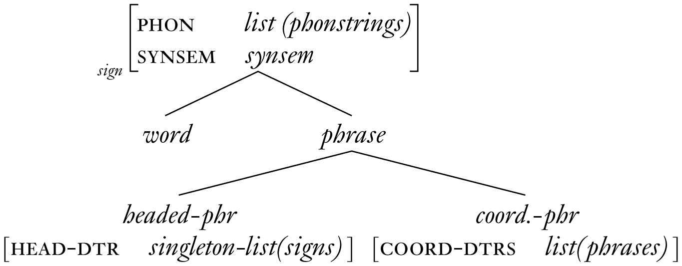
(7)



Because a type can have more than one super-type, the theory allows for multiple inheritance, which enables types to be cross-classified. For example, partitions of the sorts *constituent-structure* (i.e., head-complement, head-adjunct, head-filler, ...) and *clause-type* (i.e., declarative, interrogative, relative, ...) and their subsorts cross-classify clausal structures so that in Sag (1997), for example, a subject-relative clause like *who loves Sandy* is both a head-complement structure and a type of relative clause, and an unprefixed relative clause like *Sandy admires* is also a head-complement structure, and a different type of relative clause, while *who Sandy admires* is also a relative clause, but a head-filler phrase, rather than a head-complement phrase.

In the inheritance hierarchy in (8), words and phrases, as subsorts of *sign* have PHON values which are lists of phonstrings, and SYNSEM values which are synsems, and headed structures have a head daughter, while coordinate structures have a list of coordinate phrases as daughters.

(8)



In further partitions, the HEAD-DTR is additionally specified as a word or a phrase, and other kinds of daughter attributes may be specified. The values for the various daughter sorts are list-valued so that they can be specified as empty.

Although the theory of features in HPSG owes much to the earlier work on GPSG cited earlier, HPSG admits a larger set of value types for features. In HPSG, a feature's value belongs to one of four possible types:

- atom
- feature structure
- set of feature structures[10]
- list of feature structures[11]

If a value is not specified in a feature-structure description (i.e., an AVM), the value is still constrained by the type-definitions to be one of the possible values for that feature. That is, underspecification or nonspecification of an attribute amounts to specifying a disjunction of values allowed by the degree of underspecification. Thus, specifying either NP[NUM] or NP amounts to specifying NP[NUM *sg* ∨ *pl*], and so on, for all the possible attributes of NPs (i.e., all the features they can have).

2.2 *Constraints and structure-sharing*

An HPSG grammar consists of a set of constraints on the form of signs consistent with the constraints on the values of the features that are defined for them. Just as in GPSG, the range of possible phrase structures can be taken to be defined in terms of constraints on the well-formedness of the class of objects admitted by a particular grammar; the various phrase structure schemata that an HPSG for some language admits are in effect just very general restrictions on the combinatoric possibilities of linguistic objects. Unlike GPSG, and the theories of phrase structure which preceded it, however, HPSG does not treat constituent-structure trees as formal objects, although they remain a convenient graphic representation of the immediate constituents and linear order properties of phrasal signs. Instead, constituent structure is represented by the various DAUGHTERS attributes of phrasal signs. In informal representations, nodes are labeled by analyzable category names displayed as AVMs, and linear order is imposed.[12]

Beyond the constraints implicit in the phrase structure possiblities of the grammar, there is a variety of further restrictions. Some constraints on possible signs are inherent in the inheritance of the hierarchy of sorts. A handful of others depend on the notion of structure-sharing, explained in section 2.1.3 on types, to constrain feature-value correspondences between sisters, or between mother and some daughter, for particular features. These include

[10] Set values are represented as sequences within curly brackets: SLASH {1, 2}. The empty set is denoted: { }, while {[ ]} denotes a singleton set.

[11] List values are represented as sequences within angled brackets: SUBCAT ⟨1 [nom], 2[*inf*]⟩. The empty list is denoted: ⟨ ⟩, while ⟨[ ]⟩ denotes a singleton list.

[12] The various constituent-structure types defined by the handful of ID-Schemata that in Pollard and Sag (1994) constituted a "disjunctively specified principle of universal grammar" (1994: 38) may be considered as just a further elaboration of the type hierarchy for the type *phrase*.

CAMBRIDGE
Cambridge University Press
978-0-521-65107-3 - Studies in Contemporary Phrase Structure Grammar
Edited by Robert D. Levine and Georgia M. Green
Excerpt
More information

familiar principles like the HEAD-Feature Principle (which constrains the HEAD value of a phrase to be the same as the HEAD value of its head daughter) and the Valence Principle, as well as some form of NONLOCAL Feature Principle which governs the projection of the unbounded dependency features (SLASH, REL, and QUE).[13] Principles which constrain the CONTENT value of a phrase to be a particular function of the CONTENT values of its daughters, depending on what subtype of phrase it is, are specified in the sort declarations for particular subsorts of *phrase*.

The Valence Principle (a reformulation of the Subcategorization Principle of Pollard and Sag 1994: chs. 1–8) constrains objects of the sort *headed-phrase* so that the value of each VALENCE feature corresponds to the respective valence value of their head daughter, minus elements that correspond to values of COMP-DTRS, SUBJ-DTRS, and SPR-DTRS. In effect, the Valence Principle says that the SUBJ, COMPS and SPR values of a phrase correspond to the respective SUBJ, COMPS and SPR values of its head daughter except that the values specified for this daughter that correspond to any SUBJ-DTRS, COMPS-DTRS, and SPR-DTRS values respectively are absent from the respective valence attributes of the phrase itself.

The HEAD-Feature Principle, described above, is likewise represented in (9) as a constraint on headed phrases.

(9)
$$\begin{bmatrix} \textit{headed-phrase} \\ \text{SYNSEM}\,|\,\text{LOCAL}\,|\,\text{CATEGORY}\,|\,\text{HEAD}\ \boxed{1} \\ \text{DTR}\,|\,\text{HEAD-DTR}\ [\text{SYNSEM}\,|\,\text{LOCAL}\,|\,\text{CATEGORY}\,|\,\text{HEAD}\ \boxed{1}] \end{bmatrix}$$

Just as heads select arguments by valence features, adjuncts select heads via a HEAD feature MOD, and determiners select heads via a HEAD feature SPEC.

The use of the structure-sharing notation to express generalizations can be seen by examining a few particular cases. For example, the valence of the raising verb *tend* is represented as in (10).

(10)
$$\begin{bmatrix} \text{SUBJ}\ \boxed{1} \\ \text{COMPS}\ \text{VP}\begin{bmatrix} \text{VFORM} & \textit{inf} \\ \text{SUBJ} & \langle \boxed{1} \rangle \end{bmatrix} \end{bmatrix}$$

This constraint says that *tend* needs as a subject whatever its VP complement needs as *its* subject. It specifies *tend*'s SUBJ value as identical to the SUBJ value of the VP which *tend* selects as its complement. Similarly, (11) represents a description of the valence of a raising verb in a structure where it happens to have a quirky-case infinitive complement, as in, for example, Icelandic.

[13] Recent work suggests that each of these features may require its own respective principle regulating its propagation throughout a sign. See Sag (1997).