1
Elementary Principles of Head-Driven Phrase Structure Grammar
Georgia M. Green

1.1 Introduction

This chapter describes the theoretical foundations and descriptive mechanisms of Head-Driven Phrase Structure Grammar (HPSG), as well as proposed treatments for a number of familiar grammatical phenomena. The anticipated reader has some familiarity with syntactic phenomena and the function of a theory of syntax, but not necessarily any expertise with modern theories of phrase structure grammar. The goal of this chapter is not so much to provide a tutorial in some consistent (and inevitably dated) version of HPSG as to explicate the philosophy and techniques of HPSG grammars, and to familiarize readers with foundations and techniques of HPSG accounts of grammatical phenomena so that readers can access the primary literature.

In my opinion, the best means to fully understanding this approach, and to being able to write and read HPSG grammars, is to build an HPSG grammar from scratch, inventing and revising the details as one goes along, in accordance with the constraints imposed by the formal model (but not necessarily by every constraint ever proposed in the language of that model).

This chapter assumes the reader is curious about HPSG, perhaps attracted by claims that it aims for psychological plausibility, or that it is computationally tractable and adaptable for computational implementations in both research and practical applications, or perhaps merely interested in seeing how HPSG accounts for the properties of natural languages that any adequate theory of natural language must account for. I have sought to provide an in-depth introduction to the guiding principles and the nuts and bolts, as well to the notation, and to forgo the hard sell. Section 1.2 describes the character of HPSG grammars, and the elements and axioms of the system. Section 1.3 describes how linguistic entities are modeled,

This work was supported in part by the Beckman Institute for Advanced Science and Technology at the University of Illinois at Urbana-Champaign. Some parts are reworked versions of material that appears in Green and Levine (1999). I am grateful to Ash Asudeh, Bob Borsley, Jong-Yul Cha, Bob Levine, Carl Pollard, and Ivan Sag for comments on earlier versions, and useful advice about consistency and clarity in describing a theory that (like all theories, and maybe all organisms) evolves piecemeal, a few systems at a time.

Edited by Robert D. Borsley and Kersti Börjars.
© 2011 Blackwell Publishing Ltd. Published 2011 by Blackwell Publishing Ltd.
and how grammars describe the modeled entities. Section 1.4 describes the ontology of feature structure descriptions in HPSG, and section 1.5 deals with the expression of constraints, especially those involving the notion “same” or “matching.” Section 1.6 discusses issues relating to selection, including the treatment of agreement. Section 1.7 describes the compositional treatment of semantics in HPSG. Section 1.8 discusses the representation of constituent structure, and section 1.9 addresses the treatment of the order of elements within constituents. HPSG is very much a lexicon-driven theory, and section 1.10 describes the organization of the lexicon, relations among lexical items, and the nature of lexical rules relating them. Section 1.11 describes treatments of complementation, including the treatment of Equi and Raising constructions, and their interaction with expletive noun phrases. Section 1.12 describes variations on the treatment of so-called extraction constructions and other unbounded dependencies (e.g. pied piping), with some attention to multiple extractions and so-called parasitic gaps, as well as the nature of alleged empty categories like traces and zero pronouns. It concludes with a discussion of constraints on where extraction gaps can occur. Section 1.13 describes the HPSG account of the binding of pronouns and anaphors, and the final section indicates further directions. Two appendices summarize salient aspects of the sort inheritance hierarchies discussed, and the constraints embedded within them.

1.2 Grammars, Types, and Constraints

Two assumptions underlie the theory defining HPSGs. The first is that languages are systems of sorts of linguistic objects at a variety of levels of abstraction, not just collections of sentence(-type)s. Thus, the goal of the theory is to be able to define the grammars (or I-languages) that generate the sets of linguistic expressions (e.g. English You’ve got mail, seeks a unicorn, the,...) that represent the set of natural human languages, assigning empirically satisfactory structural descriptions and semantic interpretations, in a way that is responsive to what is known about human sentence processing. The other is that grammars are best represented as process-neutral systems of declarative constraints (as opposed to constraints defined in terms of operations on objects, as in transformational grammar). Thus, a grammar (and for that matter, a theory of Universal Grammar) is seen as consisting of an inheritance hierarchy of sorts (an is-a hierarchy), with constraints of various kinds on the sorts of linguistic object in the hierarchy. More exactly, it is a multiple-inheritance hierarchy, which simply means that sorts can inherit properties from more than one “parent.” A simple sort hierarchy can be represented as a taxonomic tree representing the sort to which belong all the linguistic entities with which the grammar deals. For each local tree in the hierarchy, the sort names that label the daughter nodes partition the sort that labels the mother; that is, they are necessarily disjoint subsorts that exhaust the sort of the mother. For example, subsorts of the sort head can be “parts of speech” (not words!) of various kinds. (Words have phonological and morphological properties, but parts of speech are abstractions, and do not.) Some of the subsorts of part-of-speech are further partitioned, as illustrated in (1).

(1) A partial inheritance hierarchy for “parts of speech”:

```
part-of-speech
  n verbal a p d ...
  v complementizer
```
A multiple-inheritance hierarchy is an interlocking set of simple hierarchies, each representing a dimension of analysis that intersects with other dimensions. The need for this sort of cross-classifying inheritance has long been obvious in the case of the lexicon: verbs have to be classified by the number and syntactic characteristics of the arguments they require, but they may also need to be classified according to inflectional class (conjugation), by semantic properties of the relations they describe (e.g. whether they represent states or properties or events, whether their subjects represent agents or experiencers, and so on (Green 1974; Levin 1993)), and of course by mood, voice, tense, and the person and number of their subjects. But comprehensive and detailed analyses of many phrasal constructions also demand the variety of perspectives that multiple inheritance reflects, as exemplified in work on inverted clauses (Green and Morgan 1996) and relative clauses (Sag 1997).

A grammar is thus a system of constraints, both unique and inherited, on sorts of linguistic objects. It would be naïve to assume that all grammars have the same sorts or the same constraints on whatever sorts they might have in common. Nevertheless, all grammars are hierarchies of sorts of phrases and words and the abstract linguistic entities that need to be invoked to define them. As detailed in appendices A and B, there are quite a few intermediate-level linguistic objects like part-of-speech that have subsorts, some of which have subsorts of their own (e.g. index, synsem, person, number, gender). All grammars constrain these various sorts in terms of properties of their component parts. One may speculate that the limited range of variation among grammars of natural languages that makes them learnable comes from the fact that grammars are as alike as they are because there are only a small number of economical solutions to the problems posed by competing forces generally present in languages. For example, languages with free word order enable subtle (non-truth-conditional) distinctions to be expressed by variation in phrase order, while languages with fixed word order simplify the task of parsing by limiting the possibilities for subsequent phrases. An elaborate inflectional system reduces ambiguity (especially temporary ambiguity), while relatively uninflected languages simplify the choices that have to be made in speech production. At the same time, whatever psychological properties and processes guide the incremental learning about the world that is universal among human beings in their first years of life must contribute to constraining grammars to be systems of information that can be learned incrementally.2

Sorts can be atomic (unanalyzed) like acc, fem, +, and sg, or they can be complex. Complex sorts of linguistic objects are defined in terms of the attributes they have (represented as features), and by the value-types of those features. In HPSG, a feature’s value may be defined to be one of four possible types:

- an atomic sort (like +, or finite);
- a feature structure of a particular sort;
- a set of feature structures;3
- a list of feature structures.4

If a value is not specified in a feature structure description, the value is still constrained by the sort-declarations to be one of the possible values for that feature. That is, it amounts to specifying a disjunction of the possible values. Thus, if the possible values for the feature num are the atomic sorts sg and pl, then specifying either NP[num] or NP amounts to specifying NP[num sg V pl], and similarly for all the other possible attributes of NPs (i.e. all the features they can have).

Sort declarations are expressed in formulae of a logic for linguistic representations (King 1989; Pollard 1999), and can be perspicuously abbreviated in labeled attribute-value matrices (AVMs) as in (2), where F1,…, Fn are feature names and sort1,…, sortk are sort names.
Sort definitions thus 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). Sorts inherit all of the attributes of their supersorts and all of the restrictions on the values of those attributes. The set of feature structures defined by a grammar is a partial subsumption ordering, that is, a transitive, reflexive, and anti-symmetric relation on the subsumption relation. (A description X is said to subsume a description Y if all of the objects described by Y are described by X. If a set is a partial subsumption ordering, then some (relevant) subsets include other (relevant) subsets.) Thus, linguistic expressions, or signs, are words or phrases, and this is reflected in the fact that the sort sign subsumes both phrase and word and no other sort. In fact, since the specifications for phrase and word are mutually exclusive (phrases have attributes that specify their immediate constituents, and words don’t), the sorts phrase and word partition the sort sign. Sorts that have no subsorts are termed “maximal sorts” because they are maximally informative or specific.

1.3 Feature Structures and Feature Structure Descriptions

All linguistic entities (including both expression types and the abstract objects that are invoked to describe them – indices, categories, cases, synsems, locals, and so on) are modeled in HPSG as feature structures. A feature structure is a complete specification of all the properties of the object it models.

To keep the distinction clear between a feature structure, which models a maximal sort, and the feature structure descriptions that are used to describe grammatically relevant classes of feature structures in the generalizations that constitute the statements of the grammar, feature structures themselves are represented as directed graphs. A partial feature structure for a simplified account of the English verb phrase sleeps is given in (3); for simplicity’s sake, the directed acyclic graph (DAG) for the non-empty-synsem-list that is the value of the two subj features is not represented.
The feature structure in (3) reflects the following information: the phrase in question has syntactic and semantic properties represented by the feature synsem, as well as the property of having a head daughter (head-dtr) but no other subconstituents; its non-hd-dtrs (non-head daughters) attribute is an empty list. Its “part of speech” (head) value is of subsort v, has finite inflectional form, and agrees with something whose agreement (agr) value is 3rd person and singular, and its head daughter’s part of speech (head) value is exactly the same. In addition, the phrase subcategorizes for (i.e. requires) a subject, but no complements, and the phrase it subcategorizes for has precisely the syntactic and semantic properties of the phrase its head daughter subcategorizes for.

As is clear from this example, the directed graphs that represent feature structures differ from the directed graphs conventionally used to represent constituent structure, in that distinct nodes can be the source for paths to (i.e. can “dominate”) a single node. This situation, as indicated by the convergence of the arrows in (3), represents the fact that the part-of-speech (of subtype v) of the head daughter is the same feature structure as the part-of-speech of the phrase itself.

Graphic representations of feature structures (like (3)) are awkward both to display and to read, so descriptions of feature structures in the form of A VMs are commonly used instead. Attribute or feature names are typically written in small capitals in A VMs, and values are written to the right of the feature name, in lower case italics if they are atomic, as in (4).

\[
\begin{array}{c}
\text{index} \\
\text{PER} & 3rd \\
\text{NUM} & sg \\
\text{GEN} & fem
\end{array}
\]

Feature structures are the entities constrained by the grammar. It is crucially important to distinguish between feature structures (fully specified objects that model linguistic expressions) and feature structure descriptions, representations (usually underspecified) that (partially) describe feature structures, and that feature structures allowed by a grammar must satisfy. 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 could not be so represented. Put another way, a partial description such as a feature structure description represented by an AVM is a constraint on members of a class of feature structures, while a total description is a constraint that 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.

### 1.4 Signs and Their Attributes

HPSGs describe languages in terms of the constraints on linguistic expressions (signs) of various types. Signs are, as in the Saussurean model, associations of form and meaning, and have two basic subsorts: phrases, which have immediate constituents; and words, which don’t. Signs are abstractions, of course; an act of uttering a linguistic expression that is modeled by a particular sign amounts to intentionally producing a sound, gesture, or graphical object that satisfies the phonological constraints on that sign, with the intent that the product of that act be understood as intended to have syntactic, semantic, and contextual properties that are modeled by the respective attributes of that sign. For more on the nature of this modeling, see Pollard and Sag (1994: 6–10, 58).
Signs have phonological, syntactico-semantic, and contextual properties, each represented by the value of a corresponding feature. Thus all signs have phon and synsem attributes, recording their phonological and syntactico-semantic structures, respectively. Phon values are usually represented in standard orthography, solely for the sake of convenience and readability. The value of the synsem attribute is a feature structure that represents the constellation of properties that can be grammatically selected for. It has a loc(al) attribute, whose value (of type local) has cat(egory), cont(ent), and context attributes. Local values are what is shared by filler and gap in so-called extraction constructions like whom you trust and Those, I watch. The synsem value also has a nonloc(al) attribute, which in effect encodes information about all types of unbounded dependency constructions (UDCs), including “missing” wh-marked subconstituents. (For the analysis of unbounded dependency constructions, see section 1.12.)

The category attribute takes as its value an entity of the sort category, whose attribute head has a part-of-speech as its value and whose subj, comps, and spr attributes have as their values lists of synsems representing the subcategorized-for arguments. The valence attributes of a sign (subj, spr, comps) record the subject, specifier, and complements that the sign subcategorizes for. These attributes take lists of synsems as their values; in S’s and referential NPs, all the lists are empty lists. These type declarations, and others discussed in this chapter, are summarized in appendix A. The partition of part of speech in (1) classifies objects having a head attribute into nouny (N, N’, NP), verby, and so on.

In the case of words, categories also have an argument structure (arg-st) feature whose value is a list of the synsems of the sign’s arguments. As with the valence features subj, comps, and spr, the synsems in the list are ordered by the obliqueness of the grammatical relations they bear, and the arg-st list represents the obliqueness record that is invoked in constraining binding relations (cf. Pollard & Sag 1994: ch. 6, or Sag et al. 2003: ch. 7). Arguments are ordered from least oblique to most oblique on the ranking familiar since Keenan and Comrie (1977): subject < direct object < secondary object < oblique argument. In most cases, the arg-st list is the concatenation of the contents of the subj, spr, and comps lists, in that order. Exceptions are provided by the pro-synsems, which represent null pronouns in null-subject languages, and the gap-synsems, which represent “extracted” elements. Both appear in arg-st lists, but not in valence lists (see section 1.11 for discussion of the latter). The null subjects of controlled infinitives appear in both arg-st lists and subj lists, though not in constituent structures (see section 1.11).

In Pollard and Sag (1994), the value of the content attribute is a nominal-object if the sign is a referring expression like them or the bagels they ate, but a parameterized state-of-affairs (or psoa) if it is a predicative expression like talks or angry or in, or a quantifier. (Psoa is not a Greek word for anything, but just a funny-looking name for a representation of propositional content as a feature structure). Psoas are subtyped by the relation they express, and have attributes for the roles of their arguments. Nominal-objects in that theory have index-valued index attributes and psoa-set valued restr(iction) attributes. More current versions (e.g. Ginzburg and Sag 2000; Sag et al. 2003; Copestake et al. 2006) of the theory include an index and a restr attribute for predicative as well as referring expressions, as illustrated in (5).

(5) a. \[
\text{INDEX} \quad \text{index} \\
\text{RESTR} \quad \{ \text{nominant-rel} \\
\quad \{ \text{NOMINATOR} \quad \text{index} \\
\quad \{ \text{NOMINEE} \quad \text{index} \} \} \}
\]
The details of this representation of propositional content do not reflect an essential property of HPSG. It would make no difference if some other kind of coherent representation of a semantic analysis were 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 of the universe of discourse, 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.

Indices for expressions that denote individuals are of the subsort $indiv-ind$, while indices for expressions that denote properties or propositions (situations) are of the sort $sit-ind$. Individual-indices in turn have attributes for $per(son)$, $num(ber)$, and $gen(der)$. For perspicuity, in abbreviated AVMs, index values are often represented as letter subscripts on category designations: NP$_i$, for example. The content specification is abbreviated as a tag following a colon after a category designation; VP$_{[I]}$ refers to a VP with the content value $[I]$.

Finally, the context attribute records indexical information (in the values of the speaker, addressee, and location features), and is supposed to represent, in the value of its background attribute, linguistically relevant information that is generally considered pragmatic. For some discussion, see Green (1995, 2000).

### 1.5 Constraints and Structure-Sharing

As indicated in section 1.3, constraints on feature structures are expressed in terms of feature structure descriptions. The more underspecified a description is, the larger the class of objects that satisfy it, and the greater the generalization it expresses. Anything that is entailed in sort definitions (including lexical representations) or in universal or language-specific constraints does not have to be explicitly mentioned in the constraints on (i.e. descriptions of) classes of linguistic objects. For example, since the (presumably universal) 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 need to be indicated only once in each representation of a phrase.

The notion of the values of two attributes being the same is modeled in feature structures as the sharing of structure. This is represented in feature structures by means of distinct paths of arcs terminating at the same node, as illustrated in (3); for this reason, this property is sometimes referred to as re-entrancy. Structure-sharing is represented in descriptions by means of identical boxed integers (tags like $[I]$) prefixed to feature structure descriptions, denoting that they are constrained to describe the same structure, as illustrated above in (5c). Technically, a tag refers to a feature structure description that unifies all of the feature structure descriptions with the same tag. The unification of two
feature structure descriptions is a consistent feature structure description that contains all of the information in each one. **Structure-sharing** is a crucial concept in HPSG. Because it refers to token-identity, and not just type-matching, it does not have a direct counterpart in transformational theories. Structure-sharing amounts to the claim that the value of some instance of an attribute is the same feature structure as the value of some other instance of an attribute, that is, it is the **same thing** – not something that just shares significant properties with it, or a different thing that happens to have all the same properties.

Because of this, the three AVMs in (6) are equivalent descriptions of the same feature structure, which in this case is a representation of a third person singular noun phrase consisting of a single head noun like *Joan* or *she*. (In the following AVMs, sort annotations are omitted, and feature-name pathways like [A [B [C x]]] are represented as [A | B | C x], as is conventional. For perspicuity, sometimes values are labeled with the name (in italics) of the sort that structures their content, but such information is usually omitted when predictable. The attributes **head-dtr** and **non-hd-dtrs** organize information about the constituent structure of phrases. Their properties are described in subsequent paragraphs and elaborated in section 1.8.)

Because identically tagged descriptions unify, all three descriptions convey exactly the same information, and there is only one way to satisfy the token-identities in the three descriptions.

Structure-sharing is a key descriptive mechanism in HPSG. For example, the structure-sharing required by the description of a topicalization structure like *whom they admired* requires the **synsem** | **local** value of the filler daughter (e.g. *whom*) to be the same as the single member of the **slash** value of the head daughter (in this case, *they admired*), as explained in section 1.12. The sort declaration in (7) constrains the properties of the mother and the daughters in a phrase of this type; it does the work of an I(mmediate)D(ominance)-rule in GPSG (an ID-schema in Pollard & Sag 1987, 1994), comparable to a phrase structure rule that does not impose linear order on the daughters.
There is a variety of restrictions that generalize across various subtypes of phrases. These are, in general, constraints on the highest type they apply to in the phrase-type hierarchy. Several depend on the notion of structure-sharing to constrain feature-value correspondences between sisters, or between mother and some daughter, for particular features. These include principles like the Head-Feature Principle (HFP, which constrains the head value of a phrase to be the same as the head value of its head daughter) and the Valence Principle (see section 1.6), as well as a principle that governs the projection of the unbounded dependency features (slash, rel, and que), such as the Nonlocal Feature Principle of Pollard and Sag (1994), or the slash Inheritance Principle and WH-Inheritance Principle of Sag (1997). Semantic compositionality principles that constrain the content value of a phrase to have a certain relation to the content values of its daughters, depending on what subtype of phrase it is, are specified in the sort declarations for high-level subsorts of phrase.

1.6 Selection and Agreement

An example of a principle that is represented as part of the description of a subsort of phrase is the Valence Principle (a reformulation of the Subcategorization Principle of Pollard and Sag 1994). It constrains subcategorization relations of every object of the sort headed-phrase so that the value of each valence feature corresponds to the respective valence value of its head daughter, minus elements that correspond to elements with the same synsem values in the non-hd-dtrs list for that phrase. In other words, 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 synsems on those lists that correspond to phrases constituting any non-head daughters are absent from the valence attributes of the mother. Thus, while wants in wants a lollipop has a singleton nouny synsem as the value of both its subj and its comps features, wants a lollipop has such a value for its subj attribute, but its comps value is the empty list, because it has an NP daughter. In versions of HPSG with defaults (such as Sag 1997 or Ginzburg & Sag 2000), the Valence Principle can be formulated as a constraint on headed phrases to the effect that the values of the valence features of a phrase are the same as those of its head daughter, except where specified to be different, as in (8). 11

There is a variety of restrictions that generalize across various subtypes of phrases. These are, in general, constraints on the highest type they apply to in the phrase-type hierarchy. Several depend on the notion of structure-sharing to constrain feature-value correspondences between sisters, or between mother and some daughter, for particular features. These include principles like the Head-Feature Principle (HFP, which constrains the head value of a phrase to be the same as the head value of its head daughter) and the Valence Principle (see section 1.6), as well as a principle that governs the projection of the unbounded dependency features (slash, rel, and que), such as the Nonlocal Feature Principle of Pollard and Sag (1994), or the slash Inheritance Principle and WH-Inheritance Principle of Sag (1997). Semantic compositionality principles that constrain the content value of a phrase to have a certain relation to the content values of its daughters, depending on what subtype of phrase it is, are specified in the sort declarations for high-level subsorts of phrase.

1.6 Selection and Agreement

An example of a principle that is represented as part of the description of a subsort of phrase is the Valence Principle (a reformulation of the Subcategorization Principle of Pollard and Sag 1994). It constrains subcategorization relations of every object of the sort headed-phrase so that the value of each valence feature corresponds to the respective valence value of its head daughter, minus elements that correspond to elements with the same synsem values in the non-hd-dtrs list for that phrase. In other words, 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 synsems on those lists that correspond to phrases constituting any non-head daughters are absent from the valence attributes of the mother. Thus, while wants in wants a lollipop has a singleton nouny synsem as the value of both its subj and its comps features, wants a lollipop has such a value for its subj attribute, but its comps value is the empty list, because it has an NP daughter. In versions of HPSG with defaults (such as Sag 1997 or Ginzburg & Sag 2000), the Valence Principle can be formulated as a constraint on headed phrases to the effect that the values of the valence features of a phrase are the same as those of its head daughter, except where specified to be different, as in (8). 11

There is a variety of restrictions that generalize across various subtypes of phrases. These are, in general, constraints on the highest type they apply to in the phrase-type hierarchy. Several depend on the notion of structure-sharing to constrain feature-value correspondences between sisters, or between mother and some daughter, for particular features. These include principles like the Head-Feature Principle (HFP, which constrains the head value of a phrase to be the same as the head value of its head daughter) and the Valence Principle (see section 1.6), as well as a principle that governs the projection of the unbounded dependency features (slash, rel, and que), such as the Nonlocal Feature Principle of Pollard and Sag (1994), or the slash Inheritance Principle and WH-Inheritance Principle of Sag (1997). Semantic compositionality principles that constrain the content value of a phrase to have a certain relation to the content values of its daughters, depending on what subtype of phrase it is, are specified in the sort declarations for high-level subsorts of phrase.

1.6 Selection and Agreement

An example of a principle that is represented as part of the description of a subsort of phrase is the Valence Principle (a reformulation of the Subcategorization Principle of Pollard and Sag 1994). It constrains subcategorization relations of every object of the sort headed-phrase so that the value of each valence feature corresponds to the respective valence value of its head daughter, minus elements that correspond to elements with the same synsem values in the non-hd-dtrs list for that phrase. In other words, 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 synsems on those lists that correspond to phrases constituting any non-head daughters are absent from the valence attributes of the mother. Thus, while wants in wants a lollipop has a singleton nouny synsem as the value of both its subj and its comps features, wants a lollipop has such a value for its subj attribute, but its comps value is the empty list, because it has an NP daughter. In versions of HPSG with defaults (such as Sag 1997 or Ginzburg & Sag 2000), the Valence Principle can be formulated as a constraint on headed phrases to the effect that the values of the valence features of a phrase are the same as those of its head daughter, except where specified to be different, as in (8). 11
Valence features of the phrase would be specified to be different in sort declarations for particular headed-phrase types only where the synsem of the signs that are sisters to the head are absent from the appropriate valence feature on the phrase, as discussed in section 1.8.

Other aspects of selection also rely on the notion of structure-sharing. Adjuncts (e.g. attributive adjuncts and temporal, locative, and manner adverbs) select heads via a head feature mod, and determiners select heads via a head feature spec in very much the same way as heads select arguments by valence features. Structure-sharing is the essence of the HFP. The HFP is described in Pollard and Sag (1994) as an independent constraint, but is perhaps more perspicuously represented as part of the sort declaration for the sort headed-phrase, as shown in (9).

\[
\begin{array}{c}
\text{SYNSEM} \\
\text{LOCAL} \\
\text{CATEGORY} \\
\text{HEAD} \\
\text{HEAD-DTR} \\
\text{SYNSEM} \\
\text{LOCAL} \\
\text{CATEGORY} \\
\text{HEAD}
\end{array}
\]

HPSG licenses phrase types either through Immediate Dominance Schemata (Pollard and Sag 1994), or through sort declarations for particular phrasal constructions (Sag 1997). Constituent–structure trees have no formal status in HPSG (as indeed they did not in the versions of transformational grammar that relied on rewriting rules), although immediate constituents are represented by the various daughters attributes of phrasal signs, and trees are used as a convenient graphic representation of the immediate constituents and linear order properties of phrasal signs. In informal arboreal representations like (10), nodes are labeled by analyzable category names in the form of AVMs (or very under-specified abbreviations for them, like NP or NP[3sg]), linear order is imposed, and branches may be annotated to indicate the relation (e.g. head, adjunct, complement) of daughter to mother. The AVMs are usually abbreviated, with predictable parts of paths suppressed as shown here.

Agreement in person, number, and gender has always (cf. Pollard & Sag 1987, 1994) been treated as a correspondence between the person, number, and gender properties of the index of a nominal expression and the (inflected) form of a word that selects it. Thus, the valence features of a verb may require, depending on its morphological type (with implications for its phonological form), that a certain argument (e.g. the subject) have certain index specifications (e.g. [per 3, num sg]), or the mod feature of an attributive adjective may require that the noun it modifies have certain number, gender, and/or case properties. Thus, wants is 3rdsg-verb and the first synsem on its arg-st list includes the specification [index [per 3rd, num sg]], and the lexical specification of these includes [mod [head n, num pl]]. Section 1.10 discusses how so-called lexical rules relate the phon value and the agreement properties of selected elements. Empirically, case concord is independent of index agreement, and this is reflected in the architecture of HPSG: index is an attribute of content, while case is a syntactic head feature.

More recently, Kathol (1999) and Bender and Flickinger (1999) have found reasons why index agreement must be mediated by an index-valued AGR feature that is a head attribute of words. Bender and Flickinger show that in the case of pronouns in tag questions like (11), the index value and the agr of a pronoun are not necessarily the same.

\[(11) \text{Everyone wins, don’t they?}\]

Kathol et al., (this volume) discuss this issue in some detail.
Pronoun antecedent agreement is a simple and direct matter of constraints requiring two nominal expressions to have structure-shared index values, under conditions imposed by the binding principles for the language. (While there are striking commonalities across languages, there is also a range of variation, so the conditions specified in the principles differ in minor details.)

For extended discussion of the issues that any approach to agreement must account for, and how they are treated in a constraint–based theory like this one, see Pollard and Sag (1994: ch. 2).

### 1.7 Semantics

As mentioned in section 1.4, every sign has a content feature that represents its semantics, and as HPSG has been committed from the start to providing an account of how the meaning of each phrase is a function of the meanings of its parts (compositionality), principles defining this function have been a part of every version of HPSG. Naturally, what this function is depends on what the representations of the semantics of the different types of phrases look like.

#### 1.7.1 The 1994 account

In Pollard and Sag (1994), the sort that is the value of the content feature varies according to whether the sign’s head value is of type noun, quantifier, or something else. For most parts
of speech, the content value is a parameterized-state-of-affairs (psoa). The term psoa is borrowed from Situation Semantics (Barwise & Perry 1983; Gawron & Peters 1990), but since HPSG is not committed to the details of Situation Semantics, psoa can be thought of as a nickname for Representation of Propositional Content. There is no reason to get hung up on the terminology. Formally, a psoa is a structure representing a semantic relation, with attributes for each of the roles defined for it. The values of those attributes are indices or psoas that are structure-shared with particular items on the arg-st list of the head. This is what links semantic roles and grammatical relations of both explicit and implied or displaced argument phrases, as illustrated in the abbreviated representation in (12). These issues are further discussed in sections 1.4 and 1.12.5.

(12)

```
word
  HEAD  v
  SUBJ  ⟨⟩
  COMPS ⟨⟩
  CONT [RESTR
    choose-rel
    AGENT [1]
    UNDERGOER [2]
```

In (12) choose-rel is the name of a subsort of psoa that represents the relation that the verb choose refers to. The immense and elaborate cross-classifying taxonomy of psoa sorts is in effect a representation of encyclopedic (world) knowledge, as seen through the eyes of a language. (See Davis 2001 for a cogent discussion of the complexity of the factors entering into the classification, and an illustration of how their interaction can be represented to reflect the correct generalizations about how semantic roles are linked to syntactic argument indices. One major achievement of this work is the multiple-inheritance hierarchy of relations according to entailments and role-types (reminiscent of Dowty’s 1991 proto-roles, but reified.) The value of content for a nominal sign is a feature structure of sort nominal-object, as shown in (13).

(13)

```
word
  CAT HEAD n
  CONT [INDEX index
    set(psoa)
```

As mentioned, indices have person, number, and gender attributes. Number and Gender values may be a function of the object in the world that is referenced by the utterance of the nominal expression (the object it is anchored to), or reflect an arbitrary property of the word, or both, depending on the language. Nominal-objects also have a restriction attribute, whose value is a set of psoas restricting the referent of the nominal-object to have certain properties. For some discussion, see Green (1995).

Quantifiers have a third kind of feature structure as their content value. As illustrated in (14), they have an attribute det, whose value is an object of sort semantic-determiner, and an attribute restind (for Restricted Index), which is a nominal-object of the subsort nonpronoun, whose index value is always a referential index (as opposed to an expletive index).
The first formulation of the compositionality principle that constrains the content of a phrase in terms of the content values of the phrase’s immediate constituents is relatively simple: the content of a phrase is structure-shared with the content of its head daughter. Because of the nature of psoa representations for all predicative expression types (phrases headed by verbs, and predicative prepositions, adjectives, and nouns), this works fine for phrases consisting of a head and complements, or a head and a subject. It doesn’t work at all for phrases consisting of a head and a modifier, like [[eats peas slowly]]. This necessitates a second, disjunctive formulation that adds the condition that if the non-head daughter is an adjunct, then the content of the phrase is structure-shared with the content of the adjunct daughter. This appears to give exactly the right results for modifiers of VPs, as long as those modifiers are analyzed like functions that take their heads as arguments, as is familiar from many semantic traditions. However, it necessitates a somewhat strained analysis of attributive adjectives.

Because the semantics of a head-adjunct phrase is the same as that of the adjunct, the content value of attributive adjectives has to have the same semantic type as nouns and NPs have (nominal object). That means they have an index attribute (whose value has to be stipulated to be the same as that of the nominal head they modify), and a restriction attribute (whose value has to be stipulated to be whatever properties are contributed by the adjective, unioned with whatever properties are contributed by the nominal expression that is modified). The treatment of quantifiers consists of a set of declarative constraints modeling the “store and retrieve” concept of “Cooper storage” (Cooper 1975, 1983). Constraints on the values of list-valued qstore and retrieved features relative to each other allow alternative interpretations of relative scope, without requiring syntactically unmotivated non-branching structure. For discussion of issues relating to the exact feature geometry for quantified expressions, see Kathol et al., (this volume).

1.7.2 Toward an improved account

It was not long before the head-adjunct clause of the Pollard and Sag (1994) Semantics Principle was shown to make empirically incorrect predictions (Kasper 1995). At the same time, work on computational implementations showed a need for semantic representations that minimized recursion. One motivation for minimizing recursion relates to the fact that in many of the natural language processing applications that utilize implementations of HPSG, it is unnecessary to resolve ambiguities of quantifier scope, as in (15).

(15) A $3000 investment is enough to become a shareholder in thousands of mutual funds.

Minimal Recursion Semantics (MRS; see Copestake et al., 2006) enables semantic representations that are underspecified with respect to quantifier scope.
Another motivation relates to the fact that computing with semantic representations with unrestricted recursion consumes inordinate quantities of computational resources. As a consequence of minimizing recursion, the content values in Sag and Wasow (1999) and Ginzburg and Sag (2000) are more uniform, and allow a simpler compositionality statement, though at the cost of additional typed indices.

In addition, the factoring entailed by the minimal recursion approach to semantics enables a feature geometry that enforces the Locality Constraint (that only immediate complements can be selected, not arguments embedded within complements); the list of semantic objects involved in a representation just has to be a sign-level attribute, rather than an attribute within content.

In MRS-style analyses, content values have three attributes: mode, index, and restriction.12 The possible values of mode are the atomic modes proposition, directive, interrogative, and reference. Signs with the mode value reference have index values that are indices to either entities (in the case that the expression is a referring expression) or situations (in the case that the expression is predicative, e.g. a verbal, adjectival, or predicative prepositional or nominal expression, like the italicized expressions in (16)).

(16) a. Kim laughed.
   b. Kim is funny.
   c. Kim is under the table.
   d. Kim is a pediatrician.

Proposition-, directive-, and interrogative-valued contents always have a situation-valued index. Restriction values are sets of typed predications, similar in structure to psoas in Pollard and Sag (1994), except that each one has a situation attribute with a sit-ind value.13 To illustrate, Kim, pediatrician, and Kim is a pediatrician in (16d) would have the content values in (17a, b, c), respectively.

(17) a. \[
\begin{align*}
\text{MODE} & \quad \text{ref} \\
\text{INDEX} & \quad 1 \\
\text{RESTR} & \quad \begin{cases}
\text{called} \\
\text{ENTITY} & \quad 1 \\
\text{NAME} & \quad \text{Kim}
\end{cases}
\end{align*}
\]

b. \[
\begin{align*}
\text{MODE} & \quad \text{prop} \\
\text{INDEX} & \quad 2 \\
\text{RESTR} & \quad \begin{cases}
\text{pediatrician} \\
\text{SIT} & \quad 2 \\
\text{INSTANCE} & \quad 1
\end{cases}
\end{align*}
\]

c. \[
\begin{align*}
\text{MODE} & \quad \text{prop} \\
\text{INDEX} & \quad 2 \\
\text{RESTR} & \quad \{1, 3\}
\end{align*}
\]

The theory as sketched and the representation in (17b) entail that it is a representation of a proposition whose index is of type situation-index, and the situation indexed is required to be described by the one-place predication that something satisfies the predi-
cate of pediatricianhood. In the representation of *Kim is a pediatrician* in (17c), that something is required to be whatever satisfies the predication in (17a) that something bears the name *Kim*.

The content value in (17c) illustrates conformity to the principles of Semantic Compositionality and Semantic Inheritance:

- **Semantic Compositionality**: A phrase’s RESTR value is the union of the RESTR values of the daughters.
- **Semantic Inheritance**: A headed-phrase’s MODE and INDEX values are structure shared with those of the head daughter.

These amount to additional constraints in the definition of *phrase* and *headed-phrase*, respectively. They are easily represented in a system where all phrases have a *dtrs* attribute whose value is a list of phrases, and headed-phrases have a *head-dtr* attribute whose value is a phrase that shares structure with one of the phrases on the *dtrs* list. For illustrative purposes, the head daughter is arbitrarily represented here as the first daughter on the *dtrs* list.

\[(18) \quad \text{Semantic Compositionality:} \]  
\[
\begin{array}{c|c|c|p{3cm}}
\text{phrase} & \text{SYNSEM} & \text{LOCAL} & \text{CONT} \\
\text{dtrs} & \{[\ldots \text{RESTR} \quad \square \ldots \square] \ldots , [\ldots \text{RESTR} \quad \square]\} \\
\end{array}
\]

\[(19) \quad \text{Semantic Inheritance:} \]  
\[
\begin{array}{c|c|c|p{3cm}}
\text{headed-phrase} & \text{SYNSEM} & \text{LOCAL} & \text{CONT} \\
\text{head-dtr} & \{[\ldots \text{MODE} \quad \square \ldots \square] \ldots , [\ldots \text{INDEX} \quad \square]\} \\
\end{array}
\]

### 1.8 Constituent Structure

As with many other aspects of grammar, HPSG allows both monotonic and default-logic accounts of constituent structure. In the monotonic account of Pollard and Sag (1994), information about the constituent structure of phrases (as well as information about the relation of the constituent parts to each other) is recorded in the various daughters attributes (*head-dtr*, *comps-dtrs*, *subj-dtr*, *filler-dtr*, *adjunct-dtr*, *spr-dtr* (*specifier-dtr*)) of particular phrase types. These features are all list-valued, enabling them to be present but empty; values for *head-dtr*, *subj-dtr*, and *spr-dtr* are limited to being no longer than singleton lists. Thus, a description like (20) indicates a phrase with three daughters: a verb head daughter, and two phrasal complement daughters (an NP and a PP).
This analysis employs an App\text{end}-synsems function that appends its second argument (a list of synsems) to a list of the synsem values of its first argument (which is a list of phrases). (The simpler Append function would not work, since the first argument would be a sign and the second argument a synsem; the value of the valence features subj, spr, and comp needs to be a list of synsems.) In the case of (20), appending the list of synsems [4] to the list of the synsems of the elements of the list [1] yields the list of synsems [4] because the list [1] is the empty list. Appending the list of synsems [5] to the list of the synsems of the elements of the list [2] yields the list of synsems [2] because the list [5] is the empty list. Appending the list of synsems [6] to the list of the synsems of the elements of the list [3] yields an empty list, because it amounts to appending the empty list [6] to the empty list of synsems of the elements of the empty list [3]. It is important to note that “NP” and “PP” are abbreviations for phrases, not synsems, since the values of subj-dtrs, comps-dtrs, etc. are lists of phrases, while the values of subj, comps, etc. are synsems.

Sag (1997) offers an alternative representation that eliminates the redundancy of daughters-features with the valence features by distinguishing subtypes of phrases in terms of relations between the values of their valence features and a non-HD-DTRS list. Considering the Valence Principle to constrain the values of the valence features of the phrases relative to their values on the head daughter and to the synsem values of the non-head daughters, as described in section 1.5, a head-subject phrase (e.g. a finite declarative clause) is defined as in (21a), and a head-complement phrase as in (21b).

(21) a. \[
\begin{align*}
&\text{hd-subj-ph} \\
&\text{SYNSEM} | \text{LOC} | \text{CAT} \quad \text{[SUBJ 〈 ]}} \\
&\text{HEAD-DTR} | \text{SYNSEM} | \text{LOC} | \text{CAT} \quad \text{[SUBJ 〈 (Ⅱ) ]}} \\
&\text{NON-HD-DTRS} \quad \text{[SYNSEM 〈 ]}}
\end{align*}
\]

b. \[
\begin{align*}
&\text{hd-comps-ph} \\
&\text{SYNSEM} | \text{LOC} | \text{CAT} \quad \text{[COMPS 〈 ]}} \\
&\text{HEAD-DTR} | \text{SYNSEM} | \text{LOC} | \text{CAT} \quad \text{[COMPS (Ⅱ) ⊕ ... ⊕ (Ⅲ) ]}} \\
&\text{NON-HD-DTRS} \quad \text{[SYNSEM 〈 ]}, ..., [SYNSEM 〈 ]}}
\end{align*}
\]
The symbol $\oplus$ represents list append function, where the list append of lists $m$ and $n$, $m \oplus n$, is the list consisting of $n$ appended to $m$. As shown in (21), some valence values of the phrase and the head daughter are required to be different, but the default valence constraint in (8) ensures that all the valence values not specified to be different will be the same. As a consequence, the analysis of the verb phrase *gives a book to Sandy* in this approach is as in (22).

\[
\begin{array}{c}
\text{PHON} \\
\text{SYNSEM} | \text{LOCAL} | \text{CAT} \\
\text{HEAD-DTR} | \text{SYNSEM} | \text{LOCAL} | \text{CAT} \\
\text{NON-HD-DTRS}
\end{array}
\]

\[
\begin{array}{c}
\text{head-comps-ph} \\
\langle \text{gives, } \{a, \text{book}\}, \{\text{to, Sandy}\} \rangle \\
\langle \text{HEAD } 4 \rangle \\
\langle \text{SUBJ } 1 \rangle \\
\langle \text{COMPS } \langle \rangle \rangle \\
\langle \text{HEAD-DTR SYNSEM LOCAL CAT} \rangle \\
\langle \text{SPR } \langle \rangle \rangle \\
\langle \text{COMPS } \langle 2, 3 \rangle \rangle \\
\langle \text{NP SYNSEM 2}, \text{ PP SYNSEM 3} \rangle
\end{array}
\]

### 1.9 Constituent Order

The general outlines of the HPSG approach to constituent order derive from the theory of linear precedence rules developed within the tradition of Generalized Phrase Structure Grammar (GPSG), as motivated in Gazdar and Pullum (1981), and summarized in Gazdar et al., (1985). There are generalizations about the order of constituents in a phrase relative to one another that standard versions of X-Bar theory (cf. Pullum 1985; Kornai & Pullum 1990) are too restrictive to capture without positing a multitude of empty nodes and forced movement chains. The theory of Linear Precedence (LP) rules allows ordering constraints to be stated in terms of any attributes of constituents, as long as the ordering relations hold of every set of sister constituents licensed by the grammar, and this proviso of Exhaustive Constant Partial Ordering (ECPO) imposes a very restrictive constraint on possible grammars. Because these ordering constraints are partial, they allow unconstrained pairs of elements to be ordered freely relative to each other. Thus, as in GPSG, so-called “free word order” (i.e. free phrase order) is a consequence of not constraining the order of constituents at all. (Genuinely free word order, where (any) words of one phrase can precede (any) words of any other phrase, requires a word-order function that allows constituents of one phrase to be interleaved with constituents of a sister phrase. See (Pullum 1982a; Pollard & Sag 1987.)

LP rules for HPSG were discussed at some length in Pollard and Sag (1987: ch. 7). It was envisioned that linear precedence constraints would be constraints on the *phon* values of phrases with content along the following lines:

- A lexical head precedes all of its sisters (or follows, depending on the language).
- Fillers precede phrasal heads.
- Less oblique complements not headed by V precede more oblique phrasal complements.

Order is represented in *phon* values because *phon* is the feature that represents the physical reflections of the constituents, which are abstract, postulated objects of type *sign*. Presumably an *Order-phon* function would apply to the list consisting of the (*Append* of the) head daughter and the list of the non-head daughters to constrain the ordering in the *phon* value of the
phrase in terms of the relevant properties of the various daughter phrases. Such a function might amount to something paraphrasable as:

In any phrase, the phon value of any filler daughter precedes the phon value of the head daughter, and the phon values of daughters that are words precede those of daughters that are phrases, etc.

As serious grammar development for a number of languages (especially notably, German and French) has made clear, word order constraints are not always compatible with the semantic and syntactic evidence for constituency. German is indisputably verb-second in main clauses and verb-final in subordinate clauses. However, the constituents of certain types of phrases may be ordered discontinuously – interleaved rather than concatenated – with sister constituents so that the position of complements (and parts of complements!) is remarkably (to English speakers, anyway) free. The German sentences glossed in (23) provide representative examples of the problem.

(23) a. Kaufen glaube ich nicht, dass Maria das Auto will.
   buy believe I not that Maria the car wants
   ‘I don’t believe Maria wants to buy the car.’

b. [Das Manuskript gezeigt] hat Maria dem Studenten.
   the manuscript shown has Maria the student-DAT
   ‘Maria has shown the manuscript to the student.’

c. Ich glaube dass der Mann das Lied hat singen können.
   I believe that the man the song has sing-INF can-INF.
   ‘I believe that the man has been able to sing the song.’

Thus, in (23a), the head of an embedded complement appears in initial position in the main clause, with its arguments and the finite verb of the complement in their canonical places. In (23b), the head of the complement and one of its arguments appears in the initial position of the main clause, with the other two arguments in their normal position after the verb. In (23c), the head (können) of the complement of the main verb of the highest embedded clause follows the finite (main) verb hat, while the complement das Lied of the more deeply embedded verb singen precedes hat.

The resolution to this dilemma constitutes a lively topic in current research. Kathol (1995), Hinrichs and Nakazawa (1999), and Reape (1994, 1996) explore these issues in more detail (see also Dowty 1996).

1.10 The Lexicon, Lexical Relations, and Lexical Rules

As in Lexical-Functional Grammar, most of the detail in individual HPSGs is encoded in the lexical entries for particular lexical elements – everything that isn’t in the (mostly) universal definitions of phrase types (which include most of the various named Principles (e.g. the HFP, the Valence Principle)). But while the specification of phrase types is hierarchical and multidimensional, the lexicon is hierarchical and multidimensional with a vengeance.

1.10.1 Organization of the lexicon

What kind of phrase will be licensed by a particular lexical head is, as described in sections 1.4–1.8, a function of the argument structure of that head (literally, of the ARG-ST value of the lexical head): what sort of arguments it needs as its complements, subject, and/or specifier,
whether any of them has a gap in it (see section 1.12), whether the subject of an infinitive complement must be the same as, or the same in reference as, some other argument of the predicate (see section 1.11). This information is to a large extent predictable: verbs of a certain valence class require at least an NP direct object; verbs of a certain subclass of that class require that NP object to be identified with the unexpressed subject of an infinitive VP complement. Third person singular present tense verbs differ systematically from other present tense verbs, and past tense and participle forms of verbs differ systematically from stem (base) forms. In addition, auxiliary verbs differ systematically from main verbs, but this distinction cross-cuts several others. Similarly, nouns are classified by whether or not they allow or require a determiner when they are singular (pronouns and proper nouns don’t allow a determiner), and what sort of determiner it can be. For example, the quantifier much goes only with a mass noun, many goes only with a plural count noun, a/an requires a singular count noun, and the and some are not selective.

Facts such as these motivate classifying the elements of the lexicon in multiple dimensions. Argument-taking predicates are classified by transitivity, by argument-coreference with a VP complement’s subject (Equi predicates), by argument-identity with a VP complement’s subject (Raising predicates), and so on. Nominal expressions are classified by properness, pronominality, and other properties. The fact that verbs of every inflected class have inflectional forms drawn from the same set of types, the fact that different forms of individual verbs encode the same semantic roles, whether or not any argument is unexpressed, and the fact that, in more-inflected languages, nouns (and adjectives) of every class have the same sets of morphological cases motivate lexical representations that link lexemes with their regular and idiosyncratic inflectional characteristics (Miller & Sag 1997; Abeillé et al. 1999; Sag & Wasow 1999). The lexeme dimension of the lexical hierarchy encodes syntactic and semantic information that distinguishes each lexical element from others, including information inherited from higher sorts, and information specifying how semantic roles are linked to grammatical relations and morphological cases (see Davis & Koenig 2000; Davis 2001), as well as any idiosyncratic syntactic or semantic information. The inflectional dimension relates to information that might be reflected in morphological inflection: for example, on a verb, the person, number, and pronominality of its arguments, as well as the presence of all arguments; on a determiner, whether it is count or mass, singular or plural, or indifferent to either distinction, and so on. Thus, there is a lexeme sort give, and whole families of words of the form give, gives, giving, gave, given.

Finally, there is no pretense that the lexicon is quite as systematic as the foregoing description makes it sound. There is no denying that some properties of lexemes can be described at the most general level only if provision is made for them to have occasional exceptions—either individual lexemes, or particular subclasses whose requirements are contrary to what is true of the class as a whole. Consequently, it is assumed that at least some specifications in the hierarchical lexicon should be represented as having default values, which is to say that inheritance within the lexicon is not strictly monotonic; default values of a sort can be contradicted in specifications for a particular subsort (including an individual lexeme). For example, the overwhelming majority of nominal lexemes are non-reflexive ([ana −]), but reflexive pronouns have to be represented as [ana +] so that the binding theory (see section 1.12) can refer to them and constrain their distribution. Thus, the specifications for the sort noun-lexeme indicate that the property of being non-reflexive is a default: the specification [ana / −] can be overridden in the specification for the reflexive pronoun subsort of pronoun-lexeme.

1.10.2 Relations among lexical entries

Three kinds of relations among lexemes motivate lexical rules, or alternatively, sets of multiple inheritances (Meurers & Minnen 1997). First, as anyone who has ever done an electronic search, compiled a concordance, or dealt with word frequency lists knows, words that have
the same stem and differ only in their inflection (e.g. for number, tense, agreement) count in one sense as “instances of the same word,” since they have the same meaning, require the same arguments filling the same roles, and so on. In another sense, of course, they are clearly not “instances of the same word,” since they have different inflections. Lexical rules allow the shared characteristics to be stated once in a relatively underspecified lexeme, with the non-shared characteristics specified by lexical rules that depend on the class (or classes) to which that lexeme belongs. (Represented declaratively, multiple inheritance allows shared lexical characteristics to be stated in a lexeme type, and cross-cutting morphosyntactic characteristics to be stated in other dimensions of the lexical hierarchy.)

Lexemes can be (somewhat less freely) related derivationally, as well as by inflection. Thus, languages typically have nouns that correspond in regular ways to verbs (e.g. agent nominalizations (do–er), zero-affix deverbal result nouns (kick, spit), and deverbal nominalizations in -tion). Languages may also have rules for verbs that correspond in regular ways to adjectives (e.g. de-adjectival causative verbs in -ify and -ize), to nouns (e.g. en-prefixed denominal verbs such as enact, empower, emplane, engulf), to zero-affixed instrumental verbs (hammer), and to change-of-state verbs (tile, bone, bottle). These relations typically embed the meaning of a lexeme with one part of speech within the meaning of a lexeme with a very general meaning, often with a different part of speech. The derived lexeme has a phonological structure that is a function of both factors. In zero-affixation cases, the matrix lexeme contributes no phonology at all. When denominal verbs get formed from nouns with irregular plurals, or deverbal nouns get formed from verbs with irregular third singular present tense forms, their categorial properties are determined by the matrix lexeme, while their inflections are completely regular. Thus, in English we have one leaf, two leaves but to leaf, it leaves; to do [du], he does [dəz] but a do [du], two dos [duz].

A third, much more restricted kind of relation involves valence alternations, where two verbs with the same phonology and roughly the same meaning map the same semantic relation to different lists of syntactic categories. Some familiar examples are:

(24) dative alternation: Kim sent Lee a letter. / Kim sent a letter to Lee.
causative alternation: Fido walked. / Kim walked Fido.
telic alternations: Dale walked. / Dale walked a mile.
Dale ate. / Dale ate lunch.

Levin (1993) offers the most complete description of English valence alternations. In the case of verb alternations, one of the alternants may have more semantic restrictions in addition to having a different mapping of arguments to roles.

1.10.3 Lexical rules

Lexical rules are an abbreviatory convenience for describing the generalizations inherent in the inheritance paths in a multiple-inheritance-hierarchy lexicon. The basic idea of a lexical rule is to describe a class of words or lexemes in terms of how its members correspond to and differ from a class of lexemes or another class of words. Thus, in general, a lexical rule says: for every lexeme meeting [such-and-such] specifications, there is a word (or a lexeme) with some particular additional constraints that satisfies all of those specifications, except for specifications that are directly or indirectly contradicted by the additional constraints. It is this last proviso that has forced the realization that sorts described in lexical rules must already exist in the lexical hierarchy, and be constrained by all the higher sorts to which they belong (cf. Bouma et al. 2000).

One obvious class of cases is the lexical rules that define inflected words on the basis of lexemes unspecified for inflections. For example, in the grammar of English, there is a
lexical rule that defines present participles in terms of verb-lexemes unspecified for any inflection, one for third singular present tense finite verbs, another for non-third singular verbs, and so on.

Other classes of lexical rules define classes of lexemes in terms of correspondences involving their valence specifications and semantic roles. For example, a causative lexical rule might define lexemes that systematically had an additional, less oblique argument with an agentive role in its semantics.

In the earlier literature (cf. Pollard & Sag 1994: ch. 9), an “extraction” lexical rule defines lexemes with a non-empty slash value and an arg-st list that systematically has one more element than the list-append of its valence-feature lists, where the local value of the additional element is the same as the slash value. Similarly, a null-pronoun lexical rule would define verb-lexemes with a pronominal element in their arg-st list that again is greater than the list-append of its valence-feature lists by exactly that element. (An alternative, explored in more recent work such as Bouma et al. 2001, and described in section 1.12.3, builds these correspondences into constraints on lexical types.) Following the analysis outlined in the last chapter of Pollard and Sag (1994) and refined in Bouma et al. (2001) and Ginzburg and Sag (2000), a Complement Extraction lexical rule defines a lexical item that is identical to a similar lexical item, except that it systematically has a synsem of type gap (as defined in (25)) on its arg-st list in some non-initial (i.e. non-subject) position, and its comps list has no corresponding synsem of any sort on it.

\[
\begin{array}{c}
gap\text{-synsem} \\
\text{LOCAL } \square \\
\text{SLASH } \{\square}\end{array}
\]

Gap-synsems never describe the syntactico-semantic properties of actual constituents, because actual constituents must be signs, and the synsem of a sign must be of the sort canonical-synsem. The constraint that a sign’s synsem must be canonical means that there are no “empty” constituents in HPSG representations of phrases.

In both analyses the theory defines words whose arg-st lists are more than the Append of their subj and comps lists, and relates them to words that are identical except for the specified properties. Other constraints (see section 1.12) require gap-synsems to share their local value with a non-local (“extracted”) dependent.

Finally, lexical rules can define both an inflection and a derived subcategorization. For example, Sag and Wasow’s (1999) passive lexical rule states a correspondence between verb lexemes whose comps list has at least one argument-saturated nominal synsem on it, and words where an argument-saturated nominal synsem with those specifications is the the sole member of the subj list, and is absent from the comps list, which may have a synsem for an (oblique) by-phrase on it whose NP object has the same index as the argument-saturated nominal synsem on the subj list of the source lexeme, as shown in (26).

\[
\begin{array}{c}
\text{verb-lexeme} \\
\text{SYNSEM LOC CAT ARG-ST } \langle\square, \{3\}, \ldots\rangle \\
\text{verb-word} \\
\text{SYNSEM LOC CAT HEAD ARG-ST } \langle\text{VFORM } psv\rangle \\
\end{array}
\]
As is well-known, non-predicative PPs like the passive by-phrase have the properties of the NPs that are their objects for many syntactic phenomena, including binding. For discussion, see Sag and Wasow (1999: ch. 7). This fact is represented in the informal notation of (26) by the coindexing of the subject NP in the domain lexeme’s valence structure with the optional prepositional phrase in the range word’s valence structure.

The informal notation in (26) represents lexical rules as functions from the class described by the domain description to the class defined by the range description. A problem with this informal notation is that it does not make explicit that all properties of the domain class are the same in the defined range class, except where specified or entailed to be different. As Meurers and Minnen (1997) show, the task of fully specifying how to compute what is defined by a lexical rule is a non-trivial one. First of all, ensuring that properties mentioned in the domain specification but not the range specification are present in the sorts defined by the rule requires a closed world assumption (Meurers & Minnen 1997). That is, it requires a conception of lexical rules as relations among already existent elements in the lexicon. It is in effect a way of describing the lexicon by organizing it, rather than, say, by using a meta-level description to generate it from “basic” elements. It is a similarly non-trivial problem to determine what properties that are mentioned in neither the domain specification nor the range specification are required (by type constraints) to be present in the defined sort. Furthermore, sometimes the properties of the resultant sort have no relation to properties of the domain sort – for example, when the values for certain features shared by both sorts are independent of each other, and when the sorts differ in such a way that features are defined for one of the sorts but not for the other. Providing a real syntax and semantics for a convenient representation for lexical rules is a non-trivial problem because the range specifications are not always consistent with the domain specifications. For instance, in the passive lexical rule in (26), constraints on word or lexeme entail that feature structures denoted by the domain description have a comps list that begins with whatever x denotes; in the range description, those same constraints entail that x be the sole member of the subj list, and that it not be on the comps list at all.

Meurers (1995) and Meurers and Minnen (1997) provide an eye-opening discussion of just what is necessary to ensure that lexical rules entail exactly what linguists expect them to entail in a fully explicit system. The latter work explains what it takes to accomplish this in a computational implementation without invoking non-branching syntactic rules in the grammar. They treat lexical rules as descriptions of parts of the sort hierarchy, licensed only if each of their subdescriptions is licensed (i.e. is defined by the grammar). At the same time, they allow for “phantom” lexical rules that are defined so that the resultant sorts cannot participate in any constructions, but can be the domain for other lexical rules. Interestingly, their computational implementation (which is faithful to the central ideas of HPSG) involves maneuvers that mimic the non-branching phrase structure rule approach, but all of that is part of the computational implementation (the compiler for the grammar), not part of the grammar.

1.11 Complementation

1.11.1 Complementizers

On the assumption that clauses and verb phrases with and without complementizers are subsorts of the same sort, (i.e. that examples like (27a, b) belong to the same (super)sort, as do examples like (28a, b, c)), HPSG analyses (e.g. Pollard & Sag 1994; Sag 1997) treat them as differing in only one or two features.
The analysis of Sag (1997) incorporates Pullum’s (1982b) insight that the English infinitival complementizer to behaves in many ways like a subject-to-subject Raising verb like seem, without claiming that to is a verb. Instead it treats verbs and complementizers generally as distinct subsorts of the part-of-speech subsort verbal. The infinitival complementizers to and for act much like Raising verbs in that they have arguments that are the same as their verbal complement’s subject. The main property that distinguishes complementizers from verbs is that the vform value of any complementizer is the same as that of its complement; since vform is a head feature, the vform value of the complementizer-headed phrase will have to be the same as the vform value of the complement also. Verbs (like know) that are indifferent to whether their finite complement has a complementizer or not (so-called “that-deletion” verbs) are simply unspecified for the category of their complement, requiring only that it have a content value of sort proposition.

Thus in Sag (1997), to is a complementizer that is like a subject-to-subject Raising verb in subcategorizing for a [vform inf] VP complement whose subject is the same as the complementizer’s subject, while for is like a Raising-to-object verb in subcategorizing for an NP complement and an inf-form CP complement whose subject is the same as the NP complement, but not subcategorizing for any subject of its own. Inverted auxiliary verbs are similarly analyzed as having no subject, but an NP complement and a VP complement whose subject shares structure with the NP complement, following analyses suggested by Borsley (1986, 1989), as discussed in Pollard and Sag (1994: 351–2).

1.11.2 Infinitival complements: Equi and Raising

Thus, infinitive complements are treated as projections of verbal heads. Equi and Raising structures both are projections of heads that subcategorize for an unsaturated predicative complement, and indeed, have the same possibilities for constituent structure – either (29a) or (29b), depending on the verb.

(29) a. \[ S \rightarrow NP \rightarrow VP \rightarrow V \rightarrow CP[\text{inf}] \]

Speaking atheoretically, the argument structure difference between Raising verbs and Equi verbs is that Raising verbs have an argument to which they don’t assign a semantic role, while Equi verbs assign roles to all their arguments. Pollard and Sag (1994) represent this difference by saying that an Equi verb subcategorizes for an NP with an index (of sort referential, i.e. not an expletive) that is the same as the index of the PRO-synsem subj specification of its complement, and assigns a semantic role to this index, as indicated in (30a), while a Raising verb requires its subject or object to share the local value of the feature structure of the synsem that its complement VP selects as subject, but assigns no semantic role to the index of that element, as indicated in (30b).

(30) a. \[ S \rightarrow NP \rightarrow VP \rightarrow V \rightarrow CP[\text{inf}] \]

b. \[ S \rightarrow NP \rightarrow VP \rightarrow V \rightarrow CP[\text{inf}] \]
The absence of a role assignment for one subcategorized element for Raising verbs entails that the content of that element has no semantic relation to the Raising verb. Thus, there is no reference in the semantic representation of *tend* in (31) to an index for *Pat*.

(31) Pat tends to like jazzy arrangements.

Assignment of a role to the index of an Equi verb’s subject entails that sentences like (32a) with active Equi complements will have different truth conditions from ones like (32b) with passive complements.

(32) a. Sandy persuaded the doctor to examine Kim.
    b. Sandy persuaded Kim to be examined by the doctor.

By the same logic, sentences with active and passive Raising complements will have the same truth-conditional semantics, as in (33).

(33) a. Sandy expected the doctor to examine Kim.
    b. Sandy expected Kim to be examined by the doctor.

The restriction that the Equi controller have a referential index follows from the assignment of a semantic role (because arguments of roles in relations have to be of sort referential; Pollard & Sag 1994: 397). This precludes the possibility of expletive Equi controllers, which indeed do not exist, although Raising controllers can have expletive subjects and complements, as illustrated in (34) and (35).

(34) a. *There tried to be a protest against garment manufacturers with plants abroad.
    b. There seemed to be a protest against garment manufacturers with plants abroad.
Structure-sharing of local values between the valence values in Raising constructions predicts the possibility of “quirky” case on Raising controllers as in Icelandic (Andrews 1982; Sag et al. 1992). That is, if a verb exceptionally has a non-nominal subject, when that verb is the head of a complement of a subject-to-subject Raising verb, the Raising verb’s subject has that non-nominate case. Structure-sharing of local values also predicts the existence of non-NP Raising controllers. Non-nominal phrases that occur as the subject of be also occur as subjects of subject-to-subject Raising verbs when the complement verb is be, as shown in (36).

(36) a. Here and not earlier seemed to be the best place to introduce that approach to extraction constructions.
   b. Grilled or baked is likely to be how they prefer their fish.
   c. Very carefully tends to be the best way to approach a 600-pound gorilla.

Semantic roles are assigned only to situational and individual indexes. Consequently, roles are never assigned to expletives, and role-assigned arguments are never expletives, but some predicates subcategorize for expletive subjects that they assign no role to, for example:

- “weather” expressions (it): rain, late, Tuesday … ;
- existential verbs (there): be, arise, occur … ;
- extraposition verbs and adjectives (it): seem, bother, obvious … .

In fact, as demonstrated by Postal and Pullum (1988), some predicative expressions subcategorize for expletive objects. For example, transitive idioms like wing, go at, out of … require an expletive object, as do object extraposition predicates like resent, take, depend upon …, which require a sentential complement in addition. The HPSG analysis is that the expletive it has a [per 3rd, num sg] index of sort it, the expletive there has a [per 3rd] index of sort there, and both index sorts are subsorts of the sort index, along with the referential subsort.

(37) Hierarchy of index subtypes

The appearance in there-constructions of agreement between the verb and its first object, as in (38), comes from the fact that the verb subcategorizes for a first complement whose num value is shared with that of its there subject.

(38) a. There are two rabbits in the garden.
   b. *There are a rabbit in the garden.

Agreement is, as usual, a linking of a morphological form of the verb to the value of the index of the subject it subcategorizes for. The Valence Principle interacts with the lexical specifications of Raising verbs to allow the subcategorization requirements of verbs recursively embedded in Raising structures, as in (39), to be satisfied by an indefinitely higher NP.
(39)  a. There seem to have to be two defenders in the backfield at all times.
   b. *There seem to have to be a keeper near the goal at all times.

Note that structure-sharing is again critical for expressing correspondences within Equi and Raising constructions. Thus, the valence specifications of the Raising verb tend are represented as in (40), as required by its being of type intr-Raising-verb, as characterized in (30b).

(40) \[
\begin{align*}
\text{SUBJ} & \quad \langle [\text{LOC} \quad \square] \rangle \\
\text{COMPS} & \quad \langle \text{VP} \left[ \text{VFORM} \quad \langle \text{inf} \quad \langle [\text{LOC} \quad \square] \rangle \rangle \right] \rangle
\end{align*}
\]

This constraint says that tend needs as a subject something with the same local values as those its VP complement needs in its subject. It specifies tend's subj value as having the same local value as the subj value of the VP that tend selects as its complement. Similarly, (41) represents a description of the valence value of a Raising verb in a structure where it happens to have a quirky-case infinitive complement.

(41) \[
\begin{align*}
\text{SUBJ} & \quad \langle [\text{LOC} \quad \square] \quad \text{NP} \rangle \\
\text{COMPS} & \quad \langle \text{VP} \left[ \text{VFORM} \quad \langle \text{inf} \quad \langle [\text{LOC} \quad \square] \quad \text{[CASE gen]} \rangle \rangle \rangle \rangle \rangle
\end{align*}
\]

The structure-sharing of subj values entails that the case of the subject selected by the VP complement must be realized on the subject of the Raising verb taking that complement.

1.12 Extraction

The general outline of the HPSG treatment of unbounded extractions follows the three-part strategy developed in GPSG (Gazdar 1981; Gazdar et al. 1985).

- An extra constituent is constrained to match a constituent missing from its clausal sister, and what is missing is represented in the description of the clausal sister as the value of the set-valued extraction-recording feature SLASH.
- The clausal sister must be missing a constituent (not necessarily an immediate constituent).
- The correspondence between the extra constituent and the constituent that is missing is recorded by means of local (mother–daughter) correspondences over an indefinitely large array of structure.

1.12.1 Licensing the “extra” constituent

Following work by Hukari and Levine (1987, 1991; Levine & Hukari 2006), HPSG distinguishes between strong and weak extractions. In strong extraction constructions like (42), the extra constituent has all the categorial properties expected for the missing constituent.
(42) a. Okra, I don’t think anyone will eat ___.
    b. The refrigerator in which everyone thinks someone left some limburger —
        may be hard to sell.

As shown in (43), the head daughter’s value for slash shares structure with the local value of a non-argument filler daughter.

\[
\begin{array}{c}
\text{head-filler phrase} \\
\text{SYNSEM | NONLOCAL | SLASH} \\
\{ \} \\
\text{HEAD-DTR | SYNSEM | NONLOCAL | SLASH} \\
\{ \} \\
\text{NON-HD-DTRS} \\
\{\{\text{SYNSEM | LOCAL | }\}} \}
\end{array}
\]

The representation in (43) is simplified for the case where there is exactly one element in the slash set of the head-daughter. The more comprehensive representation in (44) indicates that any elements in the head daughter’s value for slash that are not matched by the local value of the non-head daughter are the phrase’s value for slash.

\[
\begin{array}{c}
\text{head-filler phrase} \\
\text{SYNSEM | NONLOCAL | SLASH} \\
\{ \} \cup \{ \} \\
\text{HEAD-DTR | SYNSEM | NONLOCAL | SLASH} \\
\{ \} \cup \{ \} \\
\text{NON-HD-DTRS} \\
\{\{\text{SYNSEM | LOCAL | }\}} \}
\end{array}
\]

This allows individual gaps to be bound of independently of each other, as in multiple-gap sentences like (45).

(45) Which violins, are those sonatas, easy to play ___, on ___,?

(The symbol \( \cup \) represents disjoint set union, which is just like the familiar set union, except that it is defined only for sets with an empty intersection.)

In weak extraction phenomena, which are licensed as a property of a lexical element, a constituent that is the argument of some predicative element must be coreferential with the missing constituent. As illustrated in (46), only coindexing is required (not full categorial identity) between some argument and the value of slash on another argument.

\[
\begin{array}{c}
\text{ARG-ST} \\
\{\{\text{LOCAL | }\}} \}, \ldots , \{\text{SLASH | }\{ \} \}, \ldots 
\end{array}
\]

The sharing of structure between the value of bind and an element in the slash set of an argument ensures that the slash value remains local to the construction headed by the weak-extraction predicate, as described in section 1.12.3. The operative difference between strong and weak extractions is that in strong cases, the case value of the two elements must match; in weak ones it need not (Pollard & Sag 1994: 187). That is, in weak extractions, case is specified on arguments independently of case specified for the missing constituents, as in phrases like those in (47), where the missing constituent and the item it is coindexed with are required to have different cases.
(47)  a.  [What they will say], is hard to know ___.
       b.  He, is easy to please ___. (tough-complements)
       c.  I, am available to dance with ___. (purpose infinitives)
       d.  I gave it to the man, Dana thinks ___. is French. (bare relative clauses)

In weak extraction cases like tough-constructions, a head of the relevant class selects a complement with a non-null slash specification, as shown in (46); this entails that the complement or some descendent of it will not be lexically realized.

1.12.2 Licensing the absence of the “missing” constituent

The nature of the non-realization of the missing constituent is still a matter of some dispute. In early versions of modern phrase structure grammar, missing constituents were treated as traces, that is, as lexically defined phrasal constituents of various category types that in each case were missing a constituent of exactly that type; thus an NP-trace was NP\[slash\{NP\}], a PP-trace was PP\[slash\{PP\}], a PP\[to\]-trace was PP\[to, \slash\{PP\[to]\}\], a 3rd singular NP-trace was NP\[num sg, per 3, \slash\{NP\[num sg, per 3\]\}], and so on. Traces were licensed in phrases by defining, for each lexical element that subcategorizes for one or more phrasal sisters, a corresponding item with a phrasal trace sister. In most current versions of HPSG, this is accomplished by a lexical rule or constraint that defines lexical entries that lack certain elements on valence lists, and have corresponding elements in their slash sets.

Missing embedded subjects, which are not sisters of lexical heads, have been licensed in HPSG (following a GPSG analysis) by a lexical rule that lets a lexical head that would ordinarily subcategorize for an S instead subcategorize for a VP (i.e. an S that lacks a subject), just in case its mother is specified as missing an NP, but main clause subject relatives and interrogatives like who likes M&M’s and Who likes M&M’s? were treated as not involving any slash dependency at all, but simply a wh-valued subject. This treatment has what was at one time regarded as the happy consequence of entailing the familiar so-called that-trace facts (Gazdar et al. 1985: 57–162; Pollard & Sag 1994: 384); there is no way to generate phrases such as (48).

(48)  a.  *?the man who Bo says that ___ left
       b.  *?Which candidate did Sandy say that ___ impressed them?

However, a number of facts have more recently been seen to converge in favor of treating subject extraction as simply another instance of the same filler–gap relation as is seen in complement extraction. For example, the fact that in several languages a clause whose morphology reflects the fact that it is missing an argument is marked the same way whether it is missing a subject or a complement (Clements et al., 1983) suggests that there ought to be a universally available means of treating subject and complement extraction uniformly (Hukari & Levine 1996; Bouma et al., 2001; Ginzburg & Sag 2001: ch. 6; Levine & Hukari 2006: ch. 2). Furthermore, sentences such as (49a), which have parasitic gaps dependent on subject gaps in non-complement clauses, would not be allowed if subject extractions were described by a rule that treated missing subjects as constructions that were stipulated to consist of just an S\[slash\{NP\}] mother and an unslashed VP daughter, as the original acount maintained (Haegeman 1984; Hukari & Levine 1996; Levine & Hukari 2006: ch. 2).

(49)  a.  Sandy is someone who until you know ___ well, ___ can seem quite cold.
       b.  Sandy is someone who until you know her history, ___ can seem quite cold.
       c.  *Sandy is someone who until you know ___ well, her manner can seem quite cold.
However, such cases are predicted if all missing subjects are treated the same way as missing complements. In addition, the *that*-trace effect has been shown to vanish when material bearing a phrasal stress (such as an adverbial phrase) intervenes between the complementizer and the site of the missing subject as in (50) (a point noted in passing in Bresnan 1977 and later rediscovered and investigated by Culicover 1993).

(50) the man who they say that, after much deliberation, ___ bought the rights to all three novels

Having a Missing Dependents Constraint treat missing subjects and complements together with a single principle gives the theory of extraction constructions in HPSG a more homogeneous appearance (Bouma et al., 2001; Ginzburg & Sag 2001: ch. 6).

1.12.3 Guaranteeing that the extra constituent matches what’s missing

The correspondence between the extra constituent and the missing constituent, which precludes the possibility of sentences like (51), with an extra NP and a missing PP, is guaranteed by a constraint on the occurrence of unbounded dependency (NONLOCAL) features.

(51) *I wonder [what table]NP he will put the books [ ___ ]PP.

Such features are constrained to appear on a phrase if they are present with the same value on at least one daughter (in most recent formulations, specifications on the head daughter; cf. Sag 1997), and on a daughter constituent only if they are present with the same value on the mother. Thus, the match between the extra constituent and the missing constituent is “a global consequence of a linked series of local mother–daughter feature correspondences” (Gazdar et al. 1985: 138). In Pollard and Sag (1994), this was achieved by the Nonlocal Feature Principle, a configurational constraint on all phrases. In more recent treatments (Sag 1997; Bouma et al. 2001), a lexical constraint requires SLASH values of all elements on an ARG-ST list to be recorded in the SLASH set of the word, as shown in (52).

(52) SLASH Amalgamation Constraint:

\[
\begin{align*}
\text{word} & \\
\text{SYNSEM} & \\
\text{LOCAL} & \text{CAT} \\
\text{NONLOCAL} & \text{SLASH} \\
\text{ARG-ST} & \{\text{SLASH} \} \\
\text{BIND} & \{\} \\
\text{SLASH} & \{\} \\
\end{align*}
\]

In Bouma et al. (2001), this approach is extended to include some adverbials, by recording them as dependents in head-arguments-phrases. Having the value for SLASH be the set difference of the amalgamated set and whatever is bound off by the lexical item allows weak extractions, which are lexically governed, to be included by the constraint. The feature BIND will have a non-empty value only in the case of weak-extraction predicates, as described in section 1.12.1. A constraint on head-nexus-phrases, shown in (53), constrains the SLASH value of a phrase to be the same as the SLASH value of the head daughter.

(53) SLASH Inheritance Constraint:

\[
\begin{align*}
\text{head-nexus-ph} & \\
\text{SYNSEM} & \text{NONLOCAL} \text{SLASH} \\
\text{HEAD-DTR} & \text{SYNSEM} \text{NONLOCAL} \text{SLASH} \\
\end{align*}
\]
In strictly monotonic accounts, the nonlocal features are constrained by distinct constraints that live on different types, namely, head-filler-ph, head-nexus-ph, head-valence-ph (a subsort of head-nexus-ph that excludes head-filler-ph).

The pied piping phenomena (Ross 1967) illustrated in (54) and (55) reflect the fact that relative and interrogative properties of phrases also involve unbounded dependencies.

(54)  
- a. reports [which] they read  
- b. reports [[whose] covers] they designed  
- c. reports [the lettering [on [the covers [of [which]]]]] they designed

(55)  
- a. [[Which] reports] did they read?  
- b. [In [[which] reports]] were they quoted?  
- c. [On the covers of [which] reports]] were they pictured?

They are treated similarly to extraction dependencies being amalgamated onto heads from arguments, with rel and que values inherited from heads via a general nonlocal inheritance constraint that obviates the need for the special case in (53).

1.12.4 Multiple extractions

In contrast to many other treatments of extractions and other unbounded dependencies, in HPSG, unbounded dependency features including slash take a set of feature structures as values, not a single feature structure. This allows for the possibility of sentences with multiple (non-coreferential) extractions, as in (56).

(56)  
This is a problem which John is tough to talk to about.

Like GPSG, the account of HPSG in Pollard and Sag (1994) licenses multiple binding of a single extracted element as in (57) by not saying anything to prevent it.

(57)  
- a. That was the rebel leader who rivals of shot the British consul.
- b. Those reports, Kim filed without reading.
- c. Which relatives should we send snapshots of to?

Such structures satisfy the constraints on the occurrence of nonlocal features. There is a much wider range of acceptability judgments for such constructions than is usually acknowledged. For example, sentences like (58) are routinely cited in scholarly discussions as unacceptable (in support of a claim that gaps in non-heads are dependent on coreferential gaps in a head). In fact, however, they are perfectly acceptable to many native speakers.

(58)  
- a. That is the rebel leader who rivals of shot the British consul.
- b. Which rebel leader did you think [my talking to] would be dangerous?

After considering various alternatives, Pollard and Sag (1994) conclude that in more restrictive dialects, grammars are constrained to allow the first element on a lexical head’s argument-structure list to contain a gap only if something else on the list does.20

1.12.5 Empty categories

Neither extraction traces nor so-called null or zero pronouns need to be represented as abstract or invisible constituents. Extractions are represented as structure-sharing between a
member of a lexical head’s slash set, and the local value of an element that is on its arg-st list, but not on its comps or subj lists. The lexical rules (Pollard & Sag 1994: 446–51) that define this relation have the form of a function (schematized in (59)) from lexical entries to lexical entries that are identical except that:

a. they contain on their arg-st list an element whose local value is the same as its slash value;

b. an element with the same local value is absent from the comps list;

c. the slash set is augmented by that local value.

\[
\begin{align*}
\text{ARG-ST} & \langle \ldots, 3 \rangle \\
\text{COMPS} & \{4\} \\
\text{SLASH} & \{2\}
\end{align*}
\]

\[
\begin{align*}
\text{ARG-ST} & \langle \ldots, 4 \rangle \\
\text{SLASH} & \{3\}
\end{align*}
\]

\[
\begin{align*}
\text{ARG-ST} & \langle \ldots, 5 \rangle \\
\text{SLASH} & \{2\}
\end{align*}
\]

The symbol \( \odot \) denotes the shuffle operation on lists, defined such that the shuffle of lists \( m \) and \( n \), \( m \odot n \), is any of the lists consisting of the elements of \( m \) interleaved with those of \( n \) that retain the order defined for each list, like each shuffle of a deck of cards. Note that \( 4 \) is the same as \( 3 \) except that the slash specification is added, which entails a different tag; this formulation is consistent with the idea that specifications of the range of a lexical rule are the same as in the domain except where specified to be different.

The more recent analysis of Bouma et al. (2001) requires all words to satisfy the constraint that the list of synsems constituting the comps list is the list of synsems that is the value of arg-st minus the elements on the subj list and any gap-synsems that are on the arg-st list.\(^{21}\) In such analyses, the synsem value of *ate* in *Bagels, Kim ate* would be represented as in (60).

\[
\begin{align*}
\text{LOCAL} & \langle \rangle \\
\text{CAT} & \langle \rangle \\
\text{SUBJ} & \langle 4 \rangle \\
\text{COMPS} & \langle \rangle \\
\text{ARG-ST} & \langle 4 \rangle \\
\text{NP} & \langle 4 \rangle \\
\text{SLASH} & \{3\}
\end{align*}
\]

Null pronouns are also treated as synsems selected via a head’s argument-structure list. They have no corresponding phrases in any daughter’s specifications, and for this reason they do not appear on any valence feature list. Since they are not in a syntactic dependency relation with anything, they are not represented in slash sets either. Null pronouns are represented on the argument-structure list because they have the same binding properties as phonologically realized pronouns, and binding is a relation among elements on
argument-structure lists (see section 1.13). Because null pronouns are represented on the argument-structure list and in Content, but not on Subj or Comps lists, expressions containing them do not count as being unsaturated on that account. To illustrate, the representation of ate in an answer utterance like *Ate squid* (or the Japanese sentence *Ika-o tabeta*) is sketched in (61):

(61)

Implied arguments (e.g. the implied argument of eat in *When they’re hungry, they eat*) have been analyzed since Pollard and Sag (1987) as distinct from null pronouns. They have generic or nonspecific rather than definite referents, and are represented in the existence of appropriate roles in the Content representation. However, no indices are assigned to those roles, and there are no corresponding synsems in Arg-st and valence feature lists or slash sets. The nonspecific reference is attributable to the fact that no constraints are specified in the grammar on the index for the relevant role in the Content value, as illustrated in (62).

(62)

For discussion of the many issues involved in the analysis of extraction of and from adjuncts including examination of issues involving so-called parasitic gaps, see Kathol et al. (this volume).

1.12.6 Constraints on gaps

Following initial work by Chomsky (1964) and Ross (1967), research in syntax has sought to identify environments in which filler–gap linkages are precluded, and to formulate independent conditions that would predict them. As early as the 1970s, however, alternative explanations began to be fleshed out for the facts that the syntactic constraints were supposed to account for (e.g. Grosu 1972). More recently, it has become apparent that much of the data on which many of these constraints have been based is not representative of the syntactic classes originally assumed; acceptability often turns on the choice of lexical items or use of,
for example, a definite article rather than an indefinite article. In other cases it has become clear that phenomena used to support universal claims are in fact language-specific.

Nonetheless, certain broad classes of effects do emerge from the analysis of extraction constructions. For example, linking gaps to lexical selection predicts most of the English (generalized) “left branch” phenomena discussed by Ross and by Gazdar (1981), given that non-head left branches are not sisters of a lexical head. Similarly, the Conjunct Condition of Ross’s Coordinate Structure Constraint (precluding the extraction of only one of a set of conjuncts, as illustrated in (63a)) is also a consequence of the Nonlocal Feature Principle and the head, as illustrated in the impossibility of across-the-board extraction in (63b).

\[(63)\]
\[\begin{array}{ll}
\text{a.} & \text{*What table, did he buy } \underline{1} \text{, and two chairs?} \\
\text{b.} & \text{*What even positive integer, is four the sum of } \underline{2} \text{, and } \underline{1} \text{?}
\end{array}\]

On the other hand, the Element Condition of Ross’s Coordinate Structure Constraint, which permits only across-the-board extractions from within conjoined constituents, now seems not to be a syntactic constraint at all in the face of such acceptable sentences as (64).

\[(64)\]
\[\begin{array}{ll}
\text{a.} & \text{Concerts that short you can leave work early, hear all of } \underline{1} \text{ and get back before anyone knows you’re gone.} \\
\text{b.} & \text{Don’t tell me they drank all the whisky which I walked four miles to the store, and paid for } \underline{2} \text{ with my own money!} \\
\text{c.} & \text{There was a new episode of } \text{The Simpsons} \text{ on last Saturday, which I watched } \underline{3} \text{, and noted another bogus word.}
\end{array}\]

If there were some reason to represent the Element Condition syntactically, it would just be the addition of the boldface clause in an independently needed Coordination Principle along the lines of (65):

\[(65)\] In a coordinate structure, the category (and nonlocal) value of each conjunct daughter is an extension of that of the mother.

As for Ross’s Complex NP Constraint (which was supposed to preclude gaps in noun complements and relative clauses as in (66)), it has been known for decades that the noun-complement cases are often completely acceptable, as shown in (67).

\[(66)\]
\[\begin{array}{ll}
\text{a.} & \text{*Nelson, they quashed the report that the player choked } \underline{3} \text{.} \\
\text{b.} & \text{*Nelson, they quashed the report which Kim gave to } \underline{4} \text{.}
\end{array}\]

\[(67)\] That coach, they heard a claim that someone choked ___.

Consequently, any constraints on them are pragmatic, not syntactic, in nature. In fact, the HPSG treatment of gaps predicts that in the general case “extractions” from the clausal complements of nouns will be syntactically well-formed, since the finite clause is just a complement of the noun (fact, proposal, idea . . .), and nothing prevents extraction of arguments from complements. Constituents with relative clause sisters, on the other hand, do seem to be strictly and syntactically constrained not to contain gaps, and HPSG analyses in effect stipulate that in a relative clause, the slash value set is a singleton whose index matches that of the relative pronoun. In many analyses, subject relative clauses are analyzed as having in situ subjects, and therefore allow complement extractions, as illustrated in (68).
(68) Okra, I don’t know anyone who likes ___.

Pollard and Sag (1994) and Sag (1997) correctly predict (68) to be well-formed, since they take who here to be an in situ subject. However, Ginzburg and Sag (2001: ch. 8) and Bouma et al. (2001), following Hukari and Levine (1995, 1996), argue that even highest subject wh-phrases should be treated as extracted, attributing the special properties of (68) to the indefinite head of the apparent relative clause; with a definite head, this construction is much less acceptable:

(69) ??Okra, I don’t know the client who likes ___.

In addition, some sort of Sentential Subject Condition seems to be required to exclude gaps in clausal subjects. In fact, it is not just clausal subjects that prohibit gaps in subject position: all subject phrases headed by verbs or complementizers (verbals in Sag 1997) display the same property, and the same property holds for gerundive NPs:

(70) a. *Lou, to argue with ___ makes me sick.
   b. *Lou, that Terry argued with ___ irritated everyone.
   c. *Who do you think arguing with ___ would infuriate Terry?
   d. ?Which of the political candidates do you think that [my arguing with ___] could be productive?

How to define a natural class comprising these structures remains an open question.

1.13 Binding

The HPSG account of binding phenomena, treated in more detail in Kathol et al. (this volume), starts from the premise that theoretical inconsistencies and documented counterexamples to familiar binding theories require that sentence-internal dependencies between referential noun phrases and coreferential reflexives, reciprocals, and personal pronouns be stated in a way that does not make reference to syntactic configurations. The HPSG analysis was developed to account for the following facts:

- Anaphoric personal pronouns (“pronominals”) and reflexives (“anaphors”) are in complementary distribution, to the extent that reflexives must have a clause-mate antecedent, and pronouns may not have one.
- However:
  - Reflexives in picture-NPs can have antecedents in higher clauses. (John, thought that pictures of himself, would make a good gift.)
  - Reflexives in picture-NPs can have antecedents outside the sentence. (John, thought about the situation. Pictures of himself, would make a good gift.)
  - Reflexives with plural reference can be bound to noun phrases that jointly do not form a syntactic constituent. (Kim, told Sandy, that there were pictures of themselves, on display.)
  - Subjects of infinitives can have an antecedent in a higher clause. (The senator, arranged [for himself, to have the leading role].)

The HPSG account is framed in terms of constraints on relations between coindexed subcategorized arguments of a head (i.e. between synsem objects on an arg-str list that have
the same index value). On Pollard and Sag’s account, extraction gaps will have the same 
index-sort as any filler they are bound to, since what is structure-shared in the unbounded 
extraction dependency is a local value, and the value for index is a part of a local value. 
The HPSG account of binding is stated in terms of obliqueness-binding (o-binding), which 
is dependent on the notion of obliqueness-command (o-command), defined recursively in 
terms of the obliqueness relation that orders the synsems on an argument-structure list.

For synsem objects Y and Z, Y is less oblique than Z iff Y precedes Z on the arg-st value 
of a lexical head.

For synsem objects Y and Z with distinct local values, and Y referential, Y locally o-com-
mands Z iff

1. Y is less oblique than Z,
2. or Y locally o-commands some X that subcategorizes for Z, in the sense of Z being the 
   value for a valence feature of X.

The second clause is needed to make correct predictions in Raising and Equi constructions.

For synsem objects Y and Z, with distinct local values, Y referential, Y o-commands Z iff

1. Y is less oblique than Z,
2. or Y o-commands some X that subcategorizes for Z in the sense indicated above,
3. or Y o-commands some X whose head value is token-identical to that of Z.

Clause (3) extends o-command recursively (via clause (2)) to arguments in complements of 
an o-commanded element.

Y (locally) o-binds Z iff

- Y and Z have the same index
- and Y (locally) o-commands Z.

Z is (locally) o-free if Z is not (locally) o-bound.

Despite its non-configurational basis, the HPSG binding theory has a familiar look:

Principle A. A locally o-commanded anaphor must be locally o-bound.
Principle B. A personal pronoun must be locally o-free.
Principle C. A non-pronoun must be o-free.

Nonetheless, like the account in Reinhart and Reuland (1993), the obliqueness account dif-
ers crucially from typical configurational accounts in that it has an inherently narrower 
scope. Principle A does not constrain all anaphors to be locally o-bound (i.e. coindexed to 
something preceding them on an argument-structure list), but only those that are locally 
o-commanded (i.e. those that are non-initial on the list). This makes strong, vulnerable, and 
apparently correct claims. First, expressions that are initial elements on argument-structure 
lists are unconstrained – free to be anaphors, coindexed to anything, and vacuously satisfying 
Principle A, or to be pronouns, substantively satisfying Principle B. Thus, the theory pre-
dicts that pronominal objects in these “exempt” conditions that are coindexed to anything 
anywhere in a higher clause, or outside the sentence altogether, can be either anaphors or 
pronouns. The following kinds of phrases are thus exempt:
• pre-nominal possessives. These are determiners, with index values like NPs, but they are the first or even the unique items on the argument-structure lists of nominal heads, so they are not subject to Principle A, since they are not locally o-commanded.

(71) Bush and Dukakis charged that Noriega had contributed to each other’s campaigns.

• objects of (prepositions in) picture-NPs. These are also a unique item on an argument-structure list, and so not locally-o-commanded.

(72) a. The children thought that pictures of themselves were on sale.
    b. I suggested that portraits of themselves would amuse the twins.
    c. John knew there was a picture of himself in the post office.

• objects, when the subject is expletive. The would-be o-commander is not referential, but o-command is not defined for non-referential sorts, therefore the next item on the list is not locally o-commanded.

(73) a. They made sure that it was clear to each other why Kim had to go.
    b. John knew that there was only himself left.

• accusative subjects. As subjects, they are not locally-o-commanded. Therefore they are exempt, and can be anaphors.

(74) a. John wanted more than anything for himself to get the job.
    b. What John would prefer is for himself to get the job.

This is correct; these reflexives that contradict the naive versions of Principle A are generally replaceable with pronouns with the same reference.

Second, because non-predicative (“case-marking”) prepositions have a content value that is structure-shared with that of their object (since the preposition makes no contribution to the meaning), the prepositional phrase has a content value of the same sort as its object, and is constrained by the binding theory just as if it were an NP. Thus, in contrast to a configurational binding theory, they pose no problem; when its nominative and accusative NPs are coindexed with each other, depends on requires an anaphoric accusative and disallows a prounoun, just as trust does.

(75) a. John depends on himself to get things done.
    b. *John depends on him to get things done

(76) a. John trusts himself to get things done.
    b. *John trusts him to get things done.

Third, a pronoun or anaphor cannot have a non-pronominal “antecedent” in a lower clause because the coindexing would put the nonpronominal in violation of the HPSG Principle C.

(77) *They told him that John would get things done.
Fourth, the analysis of extraction gaps predicts that the missing element is of the same sort as the filler, and therefore predicts that (78a) is a Principle C violation, while (78b) is not.

(78) a. *John, he, said you like ____.
    b. Him, he, said you like ____.

Finally, the HPSG account of binding phenomena predicts that with multiple complements of the same verb, more oblique arguments cannot bind less oblique ones, regardless of their relative phrase order, so that (79a) and (79b) are correctly predicted to be unacceptable since the anaphor goal phrase is less oblique than the non-pronominal about-phrase.26

    b. *Marie talked to himself about John.

1.14 Further directions

HPSG has proven attractive to researchers (both scientists and engineers) seeking to harness the systematic knowledge of natural languages in applications such as automated translation assistants and natural language interfaces to a wide variety of electronically stored databases. This chapter has tried to provide an introductory survey of the domain that HPSG aims to give an account of, and the major strategies used in that endeavor.

Sag and Wasow (1999) provides a tutorial on the spirit and the mechanics of HPSG, and is accessible to anyone with a minimal background in linguistics. Sag (1997) provides a comprehensive treatment of the syntax and semantics of English relative clauses; Ginzburg and Sag (2001) gives an even more in-depth treatment of English interrogative constructions. The chapter by Kathol et al. in this volume elaborates on the properties of multiple inheritance hierarchies, and addresses problems of licensing by lexical heads, binding, extraction, linear order, and semantics in more detail than it has been possible to provide here. Levine and Green (1999) collects analyses of a variety of phenomena in English, German, and Japanese. Nerbonne et al. (1994) is exclusively HPSG analyses of a variety of phenomena in German, while Balari and Dini (1998) contains analyses of several phenomena in Romance languages, Borsley and Przepiórkowski (1999) treats phenomena in Slavic languages, and Webelhuth et al. (1999) offers detailed analyses of phenomena in both Western and non-Western languages.

Appendix A A partial summary of some basic sorts

<table>
<thead>
<tr>
<th>Sort</th>
<th>Constraints</th>
<th>Is-a</th>
</tr>
</thead>
<tbody>
<tr>
<td>sign</td>
<td>[PHON list(phonological string)]</td>
<td>ftr-str</td>
</tr>
<tr>
<td>canonical-synsem</td>
<td>SYNSEM canonical-synsem</td>
<td>synsem</td>
</tr>
<tr>
<td>gap-synsem</td>
<td>[LOCAL {} SLASH {}]</td>
<td>synsem</td>
</tr>
</tbody>
</table>

Special symbols: the sequence union of lists \[m\] and \[m \oplus n\], is the list consisting of \[n\] appended to \[m\].
Appendix A  (cont’d)

<table>
<thead>
<tr>
<th>Sort</th>
<th>Constraints</th>
<th>Is-a</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PRO-synsem</strong></td>
<td><img src="image" alt="Constraints for PRO-synsem" /></td>
<td>synsem</td>
</tr>
<tr>
<td><strong>word</strong></td>
<td><img src="image" alt="Constraints for word" /></td>
<td>sign</td>
</tr>
<tr>
<td><strong>phrase</strong></td>
<td><img src="image" alt="Constraints for phrase" /></td>
<td>sign</td>
</tr>
<tr>
<td><strong>synsem</strong></td>
<td><img src="image" alt="Constraints for synsem" /></td>
<td>ftr-str</td>
</tr>
<tr>
<td><strong>local</strong></td>
<td><img src="image" alt="Constraints for local" /></td>
<td>ftr-str</td>
</tr>
<tr>
<td><strong>nonlocal</strong></td>
<td><img src="image" alt="Constraints for nonlocal" /></td>
<td>ftr-str</td>
</tr>
<tr>
<td><strong>category</strong></td>
<td><img src="image" alt="Constraints for category" /></td>
<td>ftr-str</td>
</tr>
<tr>
<td><strong>part-of-speech</strong></td>
<td><img src="image" alt="Constraints for part-of-speech" /></td>
<td>ftr-str</td>
</tr>
<tr>
<td><strong>v</strong></td>
<td><img src="image" alt="Constraints for v" /></td>
<td>part-of-speech</td>
</tr>
<tr>
<td><strong>vform</strong></td>
<td><img src="image" alt="Constraints for vform" /></td>
<td>ftr-str</td>
</tr>
<tr>
<td><strong>n</strong></td>
<td><img src="image" alt="Constraints for n" /></td>
<td>part-of-speech</td>
</tr>
<tr>
<td><strong>fin, bse, prp, psp, psv</strong></td>
<td><img src="image" alt="Constraints for fin, bse, prp, psp, psv" /></td>
<td>vform</td>
</tr>
<tr>
<td><strong>nom, acc</strong></td>
<td><img src="image" alt="Constraints for nom, acc" /></td>
<td>case</td>
</tr>
<tr>
<td><strong>content</strong></td>
<td><img src="image" alt="Constraints for content" /></td>
<td>ftr-str</td>
</tr>
<tr>
<td><strong>context</strong></td>
<td><img src="image" alt="Constraints for context" /></td>
<td>ftr-str</td>
</tr>
<tr>
<td><strong>index</strong></td>
<td><img src="image" alt="Constraints for index" /></td>
<td>ftr-str</td>
</tr>
<tr>
<td><strong>person</strong></td>
<td><img src="image" alt="Constraints for person" /></td>
<td>ftr-str</td>
</tr>
<tr>
<td><strong>number</strong></td>
<td><img src="image" alt="Constraints for number" /></td>
<td>ftr-str</td>
</tr>
<tr>
<td><strong>gender</strong></td>
<td><img src="image" alt="Constraints for gender" /></td>
<td>ftr-str</td>
</tr>
</tbody>
</table>
### Appendix A  (cont’d)

<table>
<thead>
<tr>
<th>Sort</th>
<th>Constraints</th>
<th>Is-a</th>
</tr>
</thead>
<tbody>
<tr>
<td>gender</td>
<td>ftr-str</td>
<td></td>
</tr>
<tr>
<td>case</td>
<td>ftr-str</td>
<td></td>
</tr>
<tr>
<td>mode</td>
<td>ftr-str</td>
<td></td>
</tr>
<tr>
<td>list</td>
<td>ftr-str</td>
<td></td>
</tr>
<tr>
<td>set</td>
<td>ftr-str</td>
<td></td>
</tr>
<tr>
<td>predication</td>
<td>ftr-str</td>
<td></td>
</tr>
<tr>
<td>boolean</td>
<td>ftr-str</td>
<td></td>
</tr>
<tr>
<td>+, −</td>
<td>boolean</td>
<td></td>
</tr>
<tr>
<td>1st, 2nd, 3rd</td>
<td>person</td>
<td></td>
</tr>
<tr>
<td>sg, pl</td>
<td>number</td>
<td></td>
</tr>
<tr>
<td>fem, masc, neut</td>
<td>gender</td>
<td></td>
</tr>
<tr>
<td>prop, dir, int,</td>
<td>mode</td>
<td></td>
</tr>
<tr>
<td>reference</td>
<td>index</td>
<td></td>
</tr>
<tr>
<td>referential,</td>
<td>referential</td>
<td></td>
</tr>
<tr>
<td>expletive</td>
<td>expletive</td>
<td></td>
</tr>
<tr>
<td>indiv-ind, sit-ind</td>
<td></td>
<td></td>
</tr>
<tr>
<td>it, there</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Appendix B  Some phrasal types

Special symbols: the right-leaning slash (“/”) in a default specification has the interpretation “unless otherwise specified, has the following value.”

The symbol $\cup$ represents disjoint set union, which is just like the familiar set union, except that it is defined only for sets with an empty intersection.

The sequence difference of lists $[\mathbf{m}]$ and $[\mathbf{n}]$ $[\mathbf{m}] \mathbf{n}$ is defined as the list consisting of $[\mathbf{m}]$ minus the elements in $[\mathbf{n}]$.

<table>
<thead>
<tr>
<th>Sort</th>
<th>Constraints</th>
<th>Is-a</th>
</tr>
</thead>
<tbody>
<tr>
<td>headed-phrase</td>
<td></td>
<td>phrase</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEAD-DTR [LOCAL]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NON-HD-DTRS [</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[rest [ ]], ... ]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Appendix B  (cont’d)

<table>
<thead>
<tr>
<th>Sort</th>
<th>Constraints</th>
<th>Is-a</th>
</tr>
</thead>
<tbody>
<tr>
<td>head-nexus-ph</td>
<td>SYNSEM HEAD-DTR [SYNSEM HEAD-DTR [LOCAL CONTENT [LOCAL CONTENT [NONLOCAL</td>
<td>headed-phrase</td>
</tr>
<tr>
<td>head-comps-ph</td>
<td>HEAD-DTR word]</td>
<td>head-nexus</td>
</tr>
<tr>
<td>head-su-ph</td>
<td>SUBJ HEAD-DTR [SUBJ HEAD-DTR [LOCAL CONTENT [LOCAL CONTENT [NONLOCAL</td>
<td>head-nexus</td>
</tr>
<tr>
<td>head-spr-ph</td>
<td>SPR HEAD-DTR [SPR HEAD-DTR [LOCAL CONTENT [LOCAL CONTENT [NONLOCAL</td>
<td>head-nexus-ph</td>
</tr>
<tr>
<td>head-only-ph</td>
<td>[NON-HD-DTRS ⟨ ⟩]</td>
<td>head-nexus-ph</td>
</tr>
<tr>
<td>head-filler-ph</td>
<td>SYNSEM [NONLOCAL</td>
<td>SLASH [LOCAL v [SUBJ [COMPS [LOCAL</td>
</tr>
<tr>
<td>head-adjunct-ph</td>
<td>HEAD-DTR [SYNSEM [LOCAL [NONLOCAL</td>
<td>SLASH</td>
</tr>
<tr>
<td>word</td>
<td>ARG-ST [SUBJ [COMPS [list (gap – synsems)</td>
<td>sign</td>
</tr>
</tbody>
</table>

### Notes

2. This is the author’s speculation; see Green (1997 and this volume, chapter 11) for some discussion. Practitioners of HPSG have a range of points of view on such issues.
3. Set values are represented as sequences within curly brackets: SLASH [1, 2]. The empty set is denoted { }, while {{ }} denotes a singleton set.
4. List values are represented as sequences within angled brackets: SUBCAT ⟨NP, VP[inf]⟩. The empty list is denoted ⟨ ⟩, while ⟨ ⟨ ⟩⟩ denotes a singleton list. For convenience, accounts of HPSG often represent list and set as feature structures belonging to the highest type, with atomic subtypes elist (empty list) and nelist (non-empty list) and eset and neset, respectively. This is an oversimplification, however. Sets and lists are always sets and lists of something, so if list and set are basic types, list must have the subtypes phonstring-list,
synsem-list, phrase-list, etc., with each of these partitioned into empty and non-empty lists, and similarly for the type set. A list of synsems does not unify with a list of signs any more than non-empty sets of heads and indices could unify.

5 See section 1.5 below.


7 NB: in certain other respects the binding theory presented in Sag et al. (2003) reflects the work’s character as an introductory textbook, and does not correspond to the binding theory in Pollard and Sag (1994).

8 The tag [ ] that occurs as the value of both the index and the instance attribute in (5c) indicates that the same feature structure is the value of both attributes. This structure-sharing is discussed in detail in section 1.5.

9 See appendix A, and Ginzburg and Sag (2000) for more details.

10 These may be expressed as sort definitions for higher-level sorts. Sag (1997) is an example of this approach.

11 The right-leaning slash in a default specification has the interpretation “unless otherwise specified, has the following value.” The logic of default inheritance and the notation are described in Lascarides and Copestake (1999). If valence is an attribute of categories and its value is a feature structure with subj, comps, and spr values as described, then the Valence Principle can be very simply represented as:

```
[headed-phrase
  SYNSEM | LOCAL | CAT | VALENCE / ]
  HEAD-DTR | SYNSEM | LOCAL | CAT | VALENCE / ]
```


13 This analysis is a synthesis of the analysis of Pollard and Sag (1994) as refined in section 8.5.3 (pp. 342–3) and the MRS analysis as simplified in Sag and Wasow’s (1999) introductory-level textbook. The terminology of MRS is explained in detail in Copestake et al., (2006).

14 In a path description (i.e. in a description of a path in a feature structure from a node through attributes in complex values), “…fsd…” denotes any valid path through a feature structure satisfying the constraint fsd. In a set description, “… , fsd , … ” denotes a feature structure description that is satisfied by any feature structure that contains an arc to a node that satisfies the description fsd.

15 In many computational implementations, lexical rules are represented as non-branching (head-only or non-headed) phrase structure rules. Representing lexical rules as derivational chains in syntactic structures, with one node for each class of words involved, distorts the insight that lexical rules are intended to characterize implicational relations among classes of words in the lexicon.

16 This insight was incorporated in GPSG as the claim that to was an auxiliary verb (the unique [vform [inf]] member of a subclass of [aux +] verbs).

17 This reanalysis of complementizers eliminates the need for two “infinitival” values (base and inf). Sag (1997) and Sag et al., (2003) call the remaining value inf.

18 Most of the HPSG literature treats Raising in terms of shared synsem values. Ginzburg and Sag (2001) treat Raising in terms of shared local values, because the shared synsem value analysis incorrectly predicts that nonlocal features such as slash (or aff in French) will be represented on both the Raising verb and the verb whose logical subject is the raised NP. In addition, it appears (Sag, p.c.) that in languages where extractions are reflected in verb morphology, that morphology appears on the Raising verb, but not on the head of the complement verb.

19 The constraint was configurational because it required all nonlocal specifications on any daughters element to appear on a phrase it is an immediate constituent of, and on
at least one daughter constituent only if they are present with the same value on the mother. Specifically, in the analysis of Pollard and Sag (1994), nonlocal objects had an inherited and a to-bind attribute, which each had a nonlocal object as its value. These objects, in turn, had rel, que, and slash attributes with essentially the same sorts of set values as in more current formulations. The sole purpose of the inherited and to-bind attributes was to record the projection of the slash, rel, and que values, and to limit that projection to the phrase licensing them.

The Nonlocal Feature Principle required that the inherited value of any nonlocal feature on a phrase be the union of the inherited values for that feature on the daughters, minus the value of the to-bind feature on the head daughter. The more recent treatments described just below achieve the same effect word-internally, without invoking an inherited feature for bookkeeping. They track slash- and wh-binding in head-valence phrases through heads: Inheritance Constraints ensure that the value of each nonlocal feature on a phrase is the same as its head daughter’s value for that feature.

An alternative, based on the claim in Postal (1994) that only NPs (and not, say, PPs) are involved in so-called parasitic gaps, is to approach the analysis of these sentences in terms of coreference but not argument-identity, that is, as “null” resumptive pronouns (Sag 1997: 447–8). Levine et al. (2001) offer evidence against Postal’s claim, and in support of the Pollard and Sag (1994) analysis.

With a few verbs (e.g. eat, drink), they sometimes implicate indefinite referents referring to salient exemplars. See Cote (1996) for discussion.

See Pollard and Sag (1994: 175) and Sag (1997: 446–7) for two approaches to achieving this effect.

Cf. Goldsmith (1985) and Lakoff (1986). Postal (1998) has a syntactic account of “element extraction” that provides for the acceptability of examples like (64); Levine (2001) argues that it makes incorrect predictions so that the class of prohibited element extractions, if non-empty, is even smaller than Postal’s account predicts.

While exempt anaphors and pronouns are not freely interchangeable in discourse, the conditions favoring one over the other have to do with discourse rather than grammar, as discussed in Pollard and Sag (1994), citing Kuno (1987) and Zribi-Hertz (1989), among others.

For the same reason, this account of binding correctly predicts (i) to be acceptable, but fails to predict that (ii) is unacceptable.

(i) I talked to John about himself.
(ii) *I talked about himself to John.

References


