chapter 20

Head-Driven Phrase Structure Grammar

0. INTRODUCTION

Another major formalism for syntactic theory is ***Head-Driven Phrase Structure Grammar*** or ***HPSG***. HPSG is also a generative theory of grammar. It shares with P&P and LFG the goal of modeling how human Language is structured in the mind. HPSG and LFG in particular have many things in common. For example they both make use of a highly enriched lexicon, and the Attribute Value Matrix (AVM) notation we saw with LFG.[1]

As with our discussion of LFG, a short chapter like this cannot hope to properly cover the rich variety of work done in HPSG. In order to get a fuller picture you'll need to look at some of the primary sources of material listed in the further reading section at the end of this chapter; in particular, Sag, Wasow and Bender (2003) is a very accessible work. Another small caveat is in order before we launch into the details of the theory. For pedagogical reasons, I have couched the presentation here so that someone who has read the first 15 chapters of this book can relate the material here to what

[1] With some significant differences in notation and assumptions.

Syntax: A Generative Introduction, Third Edition. Andrew Carnie.

they already understand. Sometimes in order to do this, I've had to use metaphors and analogies that many practitioners of HPSG would disagree with. For example, I often state that some theoretical device in HPSG is the "equivalent" of something else in P&P or LFG. By this, I generally mean "does roughly the same kind of work;" I do not mean that they are necessarily notationally or empirically equivalent – as they are not.

1. FEATURES

The basic tool of linguistic description in HPSG is ***features***. There are a couple of notational systems for features; I adopt here the one used in Sag, Wasow and Bender (2003).[2] Much of the argumentation in this chapter is also taken from that book.

Features enable linguistics to talk about such information as the category of a word, what other words it must appear with (i.e., its theta grid), and what level in the tree the node is (in HPSG, bar levels are treated as features). As in LFG, features are paired with a value in an AVM, and again like LFG, features can take feature structures as values (1):

1) [AGR [NUM pl]]

The AVM in (1) says that the agreement feature for the word involves a number feature that is plural in value.

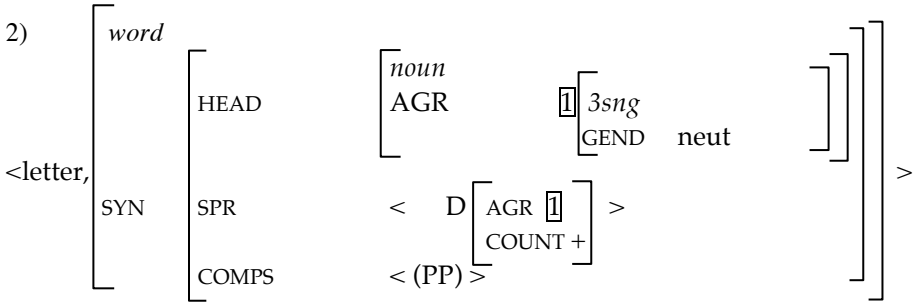
Feature structures come of a variety of types. First we have types that indicate the *word* vs. *phrase* status of every constituent in a tree (thus roughly equivalent to the notion of bar level). The features for a node are next divided into three major classes: the values of the feature SYN are structures relevant to the syntax, ARG-ST (argument structure) feature structures represent the theta grid, and the value of a SEM feature structure represents the semantic properties of the node.

Let us first talk about SYN feature structures. SYN feature structures tell us about the formal grammatical properties of the node. They tell us the syntactic category of the node, any inflectional properties of the node, what other elements the node must combine with, etc. The feature that determines the category of the node and its inflectional properties is called the HEAD feature. The feature that restricts what kind of nodes appear in the specifier position is called the SPR feature, and the feature that restricts what kind

[2] It should be noted that many of the ideas in Sag, Wasow and Bender (2003) diverge from the conception of HPSG presented in Pollard and Sag (1994); many HPSG researchers would disagree with the particulars presented here. In particular see Richter (2000) for discussion of the differences among various kinds of HPSG.

of nodes appear in the complement position is the COMPS feature. To see how this works, let us take a partial lexical entry for the word *letter*. This example is taken from Sag and Wasow (1999: 132):

2)



This looks much more intimidating than it actually is. The lexical entry is an ordered list of the form of the word and the large AVM. Ordered lists are represented with angled brackets (< >). At the top we have *word*, which tells us, obviously, that this is a word and not a phrase. Below this we have the SYN feature whose value is described by the AVM containing the HEAD, SPR, and COMPS features. The HEAD feature tells us what kind of lexical item this is. It is a *noun*, it triggers neuter and third person agreement. The SPR feature gives us an ordered list of the items that may or must appear in the specifier position of any projection of this noun. The ordered list in this lexical entry contains only one item: a count noun determiner. The boxed number found both in the HEAD feature and in this SPR feature (1) is called a ***(coreference) tag***. Tags are used to indicate that certain substructures of an overall structure are identical. The tag 1 in the description of the HEAD feature refers to the object described by the AVM that follows it. The 1 in the SPR feature indicates that whatever the agreement features of the head are, they must be identical for the specifier. This means that since the noun is third person singular, the determiner must also be singular:

3) a) this (sg) letter (sg)
   b) *these (pl) letter (sg)

The idea of structural identity (as expressed by coreference tags) does much of the work that Functional Control does in LFG and feature checking does in P&P – and more. For instance, it allows HPSG to have a non-transformational analysis of raising and to eliminate PRO in control constructions.

Notice that the specifier is not optional in this lexical entry. In English, with singular count nouns, specifiers are not optional:

4) a) I gave her the letter.
   b) I gave her a letter.
   c) *I gave her letter.

Finally, we have the COMPS feature, which says that we may have an optional PP complement:

5) a) a letter about computer software
   b) a letter from the president

The next major feature is the ARG-ST feature. Its value is an ordered list of all the arguments associated with the word and represents the theta grid of the word. You might observe that there is a redundancy between ARG-ST features, and the SPR/COMPS features. As we will see below in section 2, this is acceptable because they are treated differently by the rules which combine words into sentences. As we will see, we need the ARG-ST feature for binding reasons, independent of the SPR/COMPS features.[3] An example of the ARG-ST feature for the verb *love* is given in (6):

6) <*love*, [ARG-ST < NP, NP>]>

We can impose various kinds of selectional restrictions on this representation, of course. For example, the verb *love*s requires that its subject be third person singular. We can encode this by inserting an AVM with this specification into the first NP slot in the ARG-STR list.

7) <*loves*, [ARG-ST < [NP [AGR 3s]], NP>]>

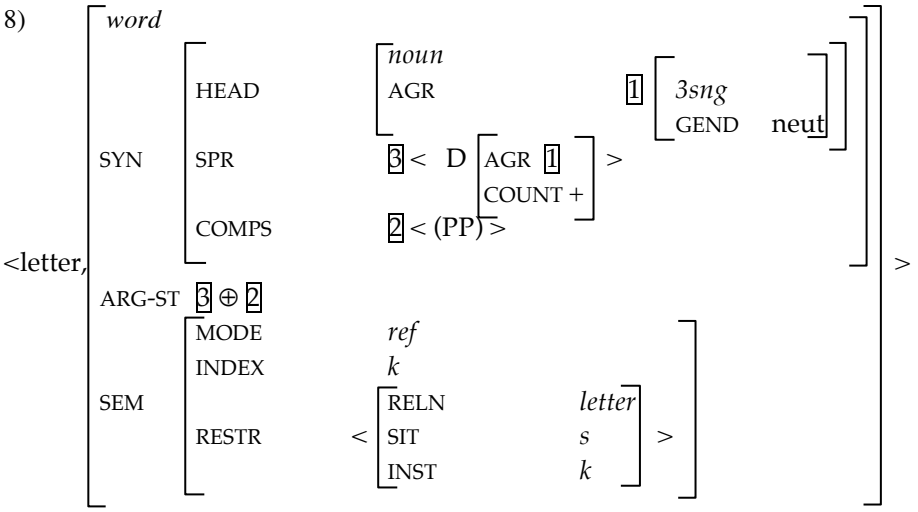
Finally we have the SEM (semantic) features. These give us information about how the word and sentence are to be interpreted. For example, MODE tells us the semantic type of the node (proposition, interrogative, directive, referential item). The INDEX features are like the indices we used in chapter 5: They mark the various participants or situations described in the sentence. Last, we have the RESTR (restriction) feature, which tell us the properties that must hold true for the sentence to be true. Again, this looks a bit like our theta grids from chapter 8. Unfortunately we don't have the space to cover this interesting aspect of HPSG in any detail.

The complete lexical entry for the noun *letter* is given in (8), showing all these features. The largest AVM in this structure is known as the noun's ***SYN-SEM structure***.

---

[3] Other arguments for distinguishing ARG-STR from COMPS and SPR can be found in Manning and Sag (1998).

8)



Each word in the lexicon has a rich lexical entry like this that specifies the features and feature structures it brings to the sentence. This lexical entry tells us that this item is a *word*; its SYN feature specifies that it is a *noun* with a 3sng, neuter HEAD feature. Since it is a count noun, the SPR feature requires that it take a count determiner, which must agree with the HEAD | AGR feature (indicated by the tag 1). It may also take an optional PP COMPS (complement). The specifier and complement features are related in the ARG-ST feature by the tags 3 and 2 (I'll discuss the ⊕ symbol below). The SEMantic features, which don't really concern us much here, indicate that the item is *ref*erential and is assigned the index *k*. The RESTR feature tells us what the word means and what context it can appear in. (Again see Sag, Wasow and Bender (2003) for a more complete description of all these features.)

2. THE LEXICON

Needless to say, there is a lot of information stored in lexical entries in HPSG grammars. In many cases this information is redundant or predictable. For example, for any count noun (such as *letter* or *ball* or *peanut*), there are two forms: a singular form and a plural form. Ideally we don't really want two complete lexical entries, as the information in one is predictable from the other. While memory is cheap, it would be nice to reduce the amount of redundancy or predictable information contained in each lexical entry. HPSG does this in a number of ways.[4]

[4] One of which we don't have the space to discuss here: *inheritance hierarchies*. These are discussed at length in Sag and Wasow (1999) and other sources on HPSG.

First, like LFG, HPSG has lexical rules that can change the shape of lexical entries. Take for example the plural rule that derives the lexical entries for plurals out of uninflected nouns. (The F_{NPL} notation refers to a function that does the actual morphology. For example, F_{NPL} applied to *cat*, will result in *cats*, but F_{NPL} applied to *child* will give you *children*.)

9) *Plural Rule*

$$\langle \boxed{1}, \begin{bmatrix} noun \\ \text{ARG-ST} \langle [\text{count } +] \rangle \end{bmatrix} \rangle \Rightarrow \langle F_{NPL}(\boxed{1}), \begin{bmatrix} word \\ \text{SYN [HEAD [AGR [NUM pl]]]} \end{bmatrix} \rangle$$

This rule says that given any basic count noun root $\boxed{1}$, you can create a plural lexical item that has an identical feature structure, except with plural number.

Similar rules can be applied to do derivation (such as deriving the noun *dancer* from the verb *dance*), or to do grammatical function-changing operations like the passive. In the following rule (taken from Sag and Wasow 1999: 235) the symbol ⊕ means "an append of lists." For more on this notion, see the references below, but for our purposes it roughly corresponds to an ordering of the AVMs. You'll see that the rule does three things: (i) it puts the verb ($\boxed{1}$), into a passive form; (ii) it puts the first argument (NP_i) into an optional *by*-phrase (don't worry about the details of the feature structure for the *by*-phrase); (iii) it moves all the other arguments ($\boxed{a}$) up in the ARG-ST list, the ⊕ symbol showing that the arguments are strictly ordered with respect to one another. The big gray arrows in the following rule are not part of the rule itself; I've just written them in to point out the important parts of the rule.

10) *Passive Rule*

$$\langle \boxed{1}, \begin{bmatrix} transitive\ verb \\ \text{ARG-ST} \langle NP_i \rangle \oplus \boxed{a} \end{bmatrix} \rangle \Rightarrow$$

(i) $\langle F_{PSP}(\boxed{1}),$ (iii)

$$\begin{bmatrix} word \\ \text{SYN} \quad [\text{HEAD [FORM pass]}] \\ \text{ARG-ST} \quad \boxed{a} \oplus \langle (\text{PP} \begin{bmatrix} \text{FORM by} \\ \text{P-OBJ} \rightarrow NP_i \end{bmatrix}) \rangle \end{bmatrix} \rangle$$

(ii)

There is one further area of redundancy in the lexicon, which we already noted in section 1 above. The information about the number of arguments a word takes appears twice in the SYN-SEM structure. It appears once in the ARG-ST feature and once in the SPR/COMPS features. This redundancy

is solved by the ***Argument Realization Principle*** (11), which builds the SPR and COMPS features out of the ARG-ST feature:

11) *Argument Realization Principle (ARP):* A word structure satisfies the following feature structure description:

$$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{SPR} & \boxed{1} \\ \text{COMPS} & \boxed{2} \end{bmatrix} \\ \text{ARG-ST} & \boxed{1} \oplus \boxed{2} \end{bmatrix}$$

The equivalence of the SPR/COMPS features and the ARG-ST feature is indicated by the tags 1 and 2. The first argument is mapped to the SPR feature, the second to the COMPS feature. This principle says that you map the first argument of the argument structure into the SPR position, and the second one into the COMPS position.

HPSG, then, shares with LFG a rich lexicon with many lexical rules. This is where the similarities end, however. In LFG, we used functions and functional equations to map constituency (c-structure) onto the functional representation (f-structure). In HPSG, like P&P theory and as we'll see below, functional (= featural) information is read directly off the constituent tree instead of using mapping principles.

3. RULES, FEATURES, AND TREES

HPSG differs from LFG in at least one significant regard: it is ***compositional***. By that we mean that the way in which the meaning of the sentence is determined is by looking at the constituent structure associated with the sentence. HPSG shares this assumption with P&P. The meaning of a sentence in HPSG and P&P can be calculated by taking each node, and using constituency to determine what modifies what. Relationships between words are directly encoded into constituency. In LFG, we instead used functional equations to calculate the relationships between words.

In all three theories (HPSG, LFG, and P&P), the end of the sentence should involve a ***saturation, satisfaction***, or ***unification*** of the features introduced by the words.[5] In Minimalism, this was accomplished by feature

[5] The exact characterization of how this works is a matter of some debate in HPSG, with various formal proposals in different versions of the theory. In early HPSG this was done with unification (see chapter 16), as is the version found in Sag and Wasow (1999). In Pollard and Sag (1994) well-formed feature structures are licensed by *conjunction* (or more accurately the conjunctive satisfaction of all the principles of the grammar). The distinctions between these approaches need not concern us here,

checking via the construction of the phrase structure tree and via movement. In LFG, this is accomplished by functional equations that map to an f-structure. In HPSG, by contrast, all feature satisfaction occurs by combining words into constituents. Constituency is introduced by phrase structure rules.[6] These rules look a little different than the ones we used in chapters 3, 6, and 7, but they do similar work. There are three basic rules (which are roughly equivalent to the complement rule, the adjunct rule and the specifier rule of X-bar theory). The first of these is the ***Head Complement Rule***. This takes a head (marked with H) word, and if it is in sequence with a number of arguments that match its COMPS requirements, licenses a phrase with an AVM like its own, except that those COMPS features have been checked off and deleted.[7]

12) *Head Complement Rule*

$$\begin{bmatrix} \textit{phrase} \\ \text{COMPS} < > \end{bmatrix} \quad \rightarrow \quad \text{H} \begin{bmatrix} \textit{word} \\ \text{COMPS} < \boxed{1}, \ldots, \boxed{n} > \end{bmatrix} \quad \boxed{1} \quad \ldots \quad \boxed{n}$$

Notice that the tags 1, …, n on the head must be identical to the tags of the phrases that follow. This means that the element(s) on the right hand side of the head must be selected for in the COMPS portion of the head word. The resulting category (the *phrase*) has had its COMPS features erased. (Thus making the satisfaction of features like COMPS very much like feature checking.) The output of this rule – when applied to a transitive verb and an NP – is seen in (13). For ease of exposition, I've omitted most of the featural information here, leaving just the relevant features present.

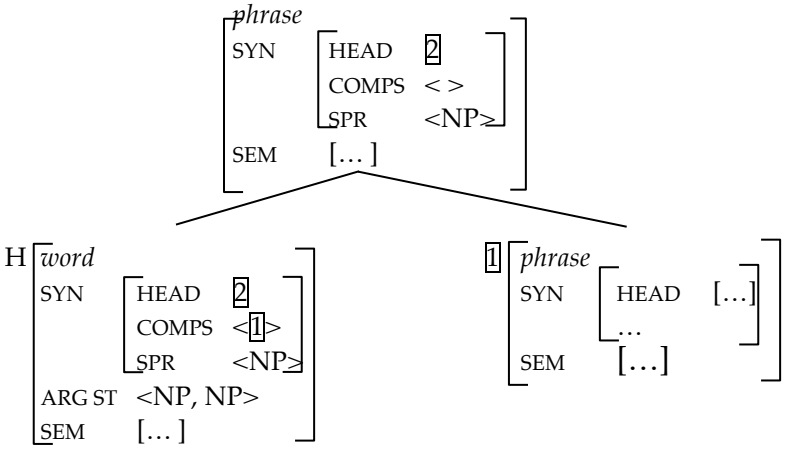
---

which is why I've adopted the neutral term *satisfaction* (or saturation) to get at the underlying idea. See Richter (2000) for a discussion of the formal apparatus underlying these issues.

[6] Phrase structure rules are actually a shorthand notation for more complicated principles. See Pollard and Sag (1994) for more discussion.

[7] For expository ease, I'm occasionally lapsing into the metaphors and terminology of Minimalism, which are not accepted by practitioners of HPSG. I do this so that you can relate the ideas of HPSG to what you have previously seen in this book; this doesn't mean that the ideas are entirely equivalent. For example, the notions of checking and deletion suggest a derivational approach to syntax which is not necessarily a part of HPSG. For more on the philosophical and methodological assumptions underlying HPSG see Pollard and Sag (1994).

13)



The resulting phrase is what we would label a V' in P&P syntax. The rule combines a head with an item that satisfies one of its COMPS requirements. It then licenses a phrase lacking that COMPS requirement.

One thing to notice about (13) is that the head features of the head word become the head features of the entire phrase, just like the syntactic category of a head is passed up the tree to the phrase level in the X-bar theory of chapter 6. In HPSG this due to the ***Head Feature Principle***:

14) *Head Feature Principle (HFP)*: The HEAD value of any headed phrase is identical to the HEAD value of the head daughter.

You'll also notice that the SPR feature of the head daughter is also transferred up to the mother node. This is triggered by the ***Valence Principle*** (Sag, Wasow and Bending 2003):

15) *Valence Principle*: Unless the rule says otherwise, the mother's SPR and COMPS values are identical to those of the head daughter.

Semantic Feature Flow

The distribution of syntactic feature values is governed by our three phrase structure rules. Semantic feature values also flow up the tree. This is governed by two principles. We won't go into these in detail, but here they are for your reference:

i) *Semantic Compositionality Principle*
In any well-formed phrase structure, the mother's RESTR value is the sum of the RESTR values of the daughters.

ii) *Semantic Inheritance Principle*
In any headed phrase, the mother's mode and index values are identical to those of the head daughter.

Adjuncts are introduced by the ***Head Modifier Rule***. This rule makes reference to the special feature MOD. The MOD feature is contained in the SYN-SEM structure of the modifier and is linked to the thing it modifies with a tag, allowing modifiers to impose selectional restrictions on the phrases they modify. The rule takes a *phrase* (equivalent to our X' level) and licenses another *phrase* (X').

16) *Head Modifier Rule*

$$[\textit{phrase}] \quad \rightarrow \quad \text{H}\,\boxed{1}\,[\textit{phrase}] \quad \begin{bmatrix} \textit{phrase} \\ \text{HEAD}\,[\text{MOD}\ \boxed{1}] \end{bmatrix}$$

Specifiers are introduced using the third rule, which also takes a *phrase* (X') and licenses a *phrase* (however, this time equivalent to our XP). The S node in HPSG is a projection of the verb (i.e., is licensed as a mother of the VP by this rule). Just as in X-bar theory, the HFP allows this rule to be non-category specific.

17) *Head Specifier Rule*

$$\begin{bmatrix} \textit{phrase} \\ \text{SPR} < > \end{bmatrix} \quad \rightarrow \quad \boxed{1} \quad \text{H} \begin{bmatrix} \textit{phrase} \\ \text{SPR} < \boxed{1} > \end{bmatrix}$$

This rule takes a phrase with a non-empty SPR value and combines it with an item that satisfies that value, and generates a phrase without an SPR value.

On an intuitive level, these rules take as inputs the lexical entries for words and output sentences where the information has been combined into a meaningful whole. In the next two sections, we turn to a variety of phenomena discussed in this book and look at how HPSG accounts for them.

4. BINDING

HPSG does not use the notion c-command to determine binding relations. Instead, binding makes reference to the ARG-ST list in the SYN-SEM structures. Because of the rules discussed above in section 3, arguments on the left side of an ARG-ST ordered list will always be higher in the tree than ones further to the right on the list. As such the binding conditions in HPSG are based on precedence in the ARG-ST list. This is accomplished with the notion ***outrank***.

18) *Outrank*: A phrase A ***outranks*** a phrase B just in the case where A's SYN-SEM structure precedes B's SYN-SEM structure on some ARG-ST list.

Anaphors are marked with a special feature: ANA. [ANA +] nodes are subject to the HPSG equivalent of principle A:

19) *Principle A:* An [ANA +] SYN-SEM structure must be outranked by a co-indexed SYN-SEM structure.

Pronouns, which are [ANA –] are subject to principle B.

20) *Principle B*: An [ANA –] SYN-SEM structure must not be outranked by a co-indexed SYN-SEM structure.

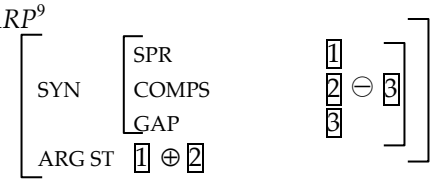
Because of the direct mapping between the ARG-ST and the tree, this will give us essentially the same results as the c-command analysis given in chapter 4. For a discussion of the important differences between a c-command analysis and an ARG-ST analysis, see Pollard and Sag (1992).

5. LONG DISTANCE DEPENDENCIES

In this section, we look briefly at how HPSG deals with long distance dependencies between the surface position of *wh*-phrases and the position with which they are thematically associated. P&P uses movement to capture this relation. LFG uses functional control. HPSG uses a feature: GAP.[8] This is a feature like COMPS or SPR that indicates that an argument is required by the SYN-SEM structure and is missing. The presence of a non-empty GAP feature indicates that an argument is not filling the expected complement position. This is encoded in a revised version of the argument realization principle, where the sequence $\boxed{2} \ominus \boxed{3}$ is a list where the elements on the list $\boxed{3}$ have been removed from $\boxed{2}$. This principle allows you to optionally map an argument to he GAP feature rather than the COMPS feature.

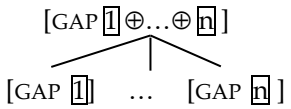
[8] Also called SLASH.

21) *Revised ARP*[9]



This guarantees that any argument that could appear on the COMPS list can appear on the GAP list instead. Just as we needed principles for passing head features, valence features and semantic features up the tree, we also need a principle to make sure GAP features make it up to the top of the tree: ***The GAP Principle***. (Formulation again taken from Sag, Wasow and Bender (2003).)

22) *The GAP Principle*



This principle encodes the idea that a mother's GAP feature represents the union of all the GAP values of its daughters.

Let's do an example. Because we don't have the space to introduce HPSG analyses of head movement or *do*-support, we'll use topicalization, rather than a *wh*-question as an example of a long distance dependency. Topicalization has the same basic properties as other *wh*-movement (subject to island constraints, etc.). The sentence we'll do is seen in (23). This sentence is grammatical if we put contrastive stress on the first NP.

23) That boy, we saw …

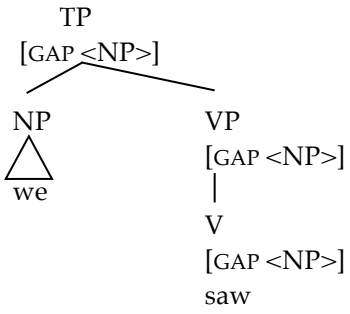
In a normal sentence, an NP (filled by *that boy*) occupies the COMPS position of the verb *saw*. In this sentence, by contrast, the COMPS position is empty. Instead there is an NP in the GAP feature. The NP *we* satisfies the verb's SPR feature. The GAP value is percolated up the tree by the GAP principle which results in a tree like (24): [10]

---

[9] This particular formulation only allows *wh*-extraction from object position. As something to think about, you might consider how HPSG would go about dealing with subject extraction (as seen in the Irish data in chapter 11).

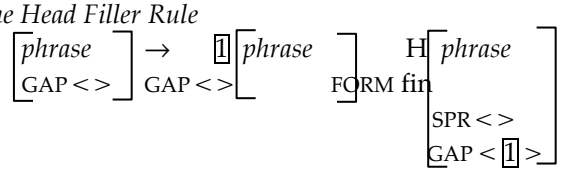
[10] You might consider whether gap feature percolation is really different from movement or is simply a notational variant. What kind of evidence might you propose to distinguish the two approaches?

24)



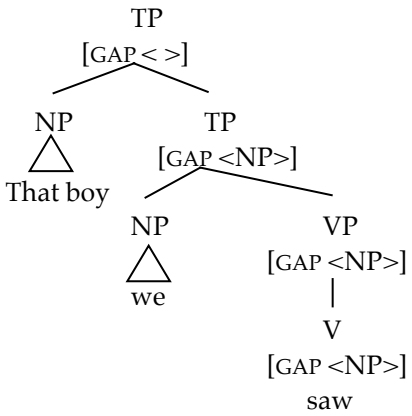
The GAP feature associated with the S node must be satisfied someway. This is accomplished with the ***Head Filler Rule***.

25) *The Head Filler Rule*



This rule satisfies the GAP feature, by adding the missing NP at the top of the tree:

26)



You can now try GPS 1 and CPS 1 & 2

IDEAS, RULES, AND CONSTRAINTS INTRODUCED IN THIS CHAPTER

Some of the definitions here are taken from Sag, Wasow and Bender (2003) or the first edition of that book (Sag and Wasow 1999).

i) ***Features***: These do the work of determining what can combine with what.
   a) Bar-level-like features tell us what hierarchical level the node is at.
   b) The SYN feature structure gives us the syntactic info about the node.
      i) The HEAD feature gives the category and inflectional info.
      ii) The COMPS feature tells us what complements appear in the structure.
      iii) The SPR feature tells us what appears in the specifier.
      iv) The GAP feature tells us if there is a long distance dependency.
   c) The ARG-ST feature is the HPSG equivalent of the theta grid. Binding relations are defined against this.
   d) The SEM feature structures tell us the semantic information about the constituent, and come in a variety of types.

ii) ***(Coreference) Tags***: Numbers written in boxes (e.g., 1) that show that two items are identical in a SYN-SEM structure or between SYN-SEM structures.

iii) ***SYN-SEM Structure***: The set of AVMs for a node, containing all the SYN, SEM and ARG-ST features.

iv) ***Plural Rule***:

$$\left\langle \boxed{1}, \begin{bmatrix} noun \\ \text{ARG-ST} \langle [\text{count } +] \rangle \end{bmatrix} \right\rangle \Rightarrow \left\langle F_{NPL}(\boxed{1}), \begin{bmatrix} word \\ \text{SYN [HEAD [AGR [NUM pl]]]} \end{bmatrix} \right\rangle$$

v) ***Passive Rule***:

$$\left\langle \boxed{1}, \begin{bmatrix} transitive\ verb \\ \text{ARG-ST} \langle NP_i \rangle \oplus \boxed{a} \end{bmatrix} \right\rangle \Rightarrow$$

$$\left\langle F_{PSP}(\boxed{1}), \begin{bmatrix} word \\ \text{SYN} \quad [\text{HEAD [FORM pass]}] \\ \text{ARG-ST} \quad \boxed{a} \oplus \langle (\text{PP} \begin{bmatrix} \text{FORM by} \\ \text{P-OBJ } NP_i \end{bmatrix}) \rangle \end{bmatrix} \right\rangle$$

vi) ***Argument Realization Principle (ARP)*** *(revised)*:

$$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{SPR} & \boxed{1} \\ \text{COMPS} & \boxed{2} \ominus \boxed{3} \\ \text{GAP} & \boxed{3} \end{bmatrix} \\ \text{ARG ST} & \boxed{1} \oplus \boxed{2} \end{bmatrix}$$

vii) ***Compositional***: The idea that the semantics of the sentence can be read off of the constituency tree. This idea is shared by P&P and HPSG, but is rejected by LFG.

viii) ***Feature Satisfaction*** *(sometimes loosely called* ***Unification****)*: The idea that all the features in a SYN-SEM structure must match. The rough equivalent of feature checking in P&P/Minimalism.

ix) ***Head Complement Rule***:

$$\begin{bmatrix} \textit{phrase} \\ \text{COMPS} < > \end{bmatrix} \rightarrow \text{H} \begin{bmatrix} \textit{word} \\ \text{COMPS} < \boxed{1}, \ldots, \boxed{n} > \end{bmatrix} \; \boxed{1} \; \ldots \; \boxed{n}$$

x) ***Head Feature Principle***: The HEAD value of any headed phrase is identical to the HEAD value of the head daughter.

xi) ***Valence Principle***: Unless the rule says otherwise, the mother's SPR and COMPS values are identical to those of the head daughter.

xii) ***Semantic Compositionality Principle***: In any well-formed phrase structure, the mother's RESTR value is the sum of the RESTR values of the daughters.

xiii) ***Semantic Inheritance Principle***: In any headed phrase, the mother's mode and index values are identical to those of the head daughter.

xiv) ***Head Modifier Rule***:

$$[\textit{phrase}] \rightarrow \text{H} \; \boxed{1} \; [\textit{phrase}] \quad \begin{bmatrix} \textit{phrase} \\ \text{HEAD} \; [\text{MOD} \; \boxed{1}] \end{bmatrix}$$

xv) ***Head Specifier Rule***:

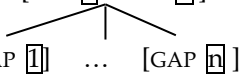
$$\begin{bmatrix} \textit{phrase} \\ \text{SPR} < > \end{bmatrix} \rightarrow \boxed{1} \quad \text{H} \begin{bmatrix} \textit{phrase} \\ \text{SPR} < \boxed{1} > \end{bmatrix}$$

xvi) ***Outrank***: A phrase A outranks a phrase B just in the case where A's SYN-SEM structure precedes B's SYN-SEM structure on some ARG-ST list.

xvii) ***Principle A***: An [ANA +] SYN-SEM structure must be outranked by a coindexed SYN-SEM structure.

xviii) ***Principle B***: An [ANA –] SYN-SEM structure must not be outranked by a coindexed SYN-SEM structure.

xix) ***The GAP Principle***: [GAP $\boxed{1} \oplus \ldots \oplus \boxed{n}$]



xx) ***The Head Filler Rule***:

$$\begin{bmatrix} \textit{phrase} \\ \text{GAP} < > \end{bmatrix} \rightarrow \boxed{1} \begin{bmatrix} \textit{phrase} \\ \text{GAP} < > \end{bmatrix} \quad \text{H} \begin{bmatrix} \textit{phrase} \\ \text{FORM fin} \\ \text{SPR} < > \\ \text{GAP} < \boxed{1} > \end{bmatrix}$$

FURTHER READING:

Borsley, Robert (1996) *Modern Phrase Structure Grammar*. Oxford: Blackwell.

Gazdar, Gerald, Ewan Klein, Geoffrey Pullum and Ivan Sag (1985) *Generalized Phrase Structure Grammar*. Cambridge: Harvard University Press.

Manning, Christopher and Ivan Sag (1998) Argument structure, valence and binding. *The Nordic Journal of Linguistics* 21, 107-44.

Pollard, Carl and Ivan Sag (1992) Anaphors in English and the scope of Binding Theory. *Linguistic Inquiry* 23, 261-303.

Pollard, Carl and Ivan Sag (1994) *Head-Driven Phrase Structure Grammar*. Stanford: CSLI Publications and Chicago: The University of Chicago Press.

Richter, Frank (2000) A Mathematical Formalism for Linguistic Theories with an Application in Head-Driven Phrase Structure Grammar. Ph.D. dissertation, University of Tübingen.

Sag, Ivan and Thomas Wasow (1999) *Syntactic Theory: A Formal Introduction* (1st ed.). Stanford: CSLI Publications.

Sag, Ivan, Thomas Wasow and Emily Bender (2003) *Syntactic Theory: A Formal Introduction* (2nd ed.). Stanford: CSLI Publications.

GENERAL PROBLEM SET

1. ENGLISH

[Application of Skills; Intermediate]

Create an HPSG-style lexicon for the words in (a) and then draw a tree using the rules and lexical entries for the sentence (b).

a) the, kitten, tore, toilet, paper
b) The kitten tore the toilet paper.

You may abbreviate your SYN-SEM structures using tags.

CHALLENGE PROBLEM SETS

CHALLENGE PROBLEM SET 1: SUBJECT-AUX INVERSION

[Critical Thinking; Challenge]

How might HPSG go about doing subject-aux inversion? (Hint: consider a lexical rule.) Assume that auxiliary verbs select for other verbs in their ARG-ST features.

CHALLENGE PROBLEM SET 2: ISLAND CONSTRAINTS

[Critical Thinking; Challenge]

In this chapter, we didn't talk at all about how HPSG might account for island constraints. Propose a constraint on the GAP principle that might account for them.