A Basic Overview of HPSG[1]

1. Overview of the framework

Head-Driven Phrase Structure Grammar (HPSG) is a generative grammar which shares with P&P[2] the common goal of building a scientific theory of the knowledge in the mind of the speaker that makes language possible (Pollard, 1997: 1). Kim (2000: 7) describes it as 'a non-derivational, constraint-based, surface oriented grammatical architecture'. By non-derivational, it is meant that HPSG has no notion of deriving one structure or representation from another, as, for example, through transformations or operations such move-a in P&P (Kim, 2000: 7-8). Instead, different representations are just subparts of a single larger structure related by declarative[3] constraints (Pollard, 1997: 5) (thus, it is constraint based). It is said to be surface oriented because it provides a direct characterization of the surface order of elements in a sentence (Shieber, 1986: 6-7). According to Shieber, this last characteristic is an assumed requirement of unification-based formalisms, which include HPSG because it uses an operation called unification (described in more detail below). Another major characteristic of HPSG is that it is highly lexicalist in that it makes use of a rich and complex lexicon in its representations (Kim, 2000: 8). A brief illustration of these properties of HPSG and a sketch of how it works will be presented in section 2.

HPSG has been greatly influenced by the Generalized Phrase Structure Grammar (GPSG). HPSG and GPSG have a number of points in common, such as the fact that both are unification-based (Shieber, 1986) and the assumption that sentences have one level of syntactic structure (Borsley, 1996: 2-3). HPSG is seen as a later development of GPSG (e.g. Sells, 1985:133 and Shieber, 1986: 53), but it is worth noting that HPSG has had influences from a number of linguistic theories (Cooper, 1996: 191), such as

Functional Unification Grammar, Lexical Functional Grammar (Shieber, 1986: 54) and P&P (Carnie, 2002: 357 and Borsley, 1996: 7). HPSG and GPSG differ in important aspects. For example, HPSG categories are more complex than those in GPSG (Borsley, 1996: 31-38) and HPSG makes more specific claims about universals and variation than the more conservative GPSG (Carnie, 2002: 357).

HPSG puts a lot of emphasis on the precise mathematical modelling of linguistic entities. Because of its focus on precision, a lot of linguistic computer implementations are based in HPSG (Borsley, 1991: 210). Computer science, together with related areas such as logic, set-theory algebra and graph theory, also influenced HPSG, as linguists working with this framework are willing to learn about these areas in order to make their analyses more precise (Pollard, 1997: 2). However, as Borsley (1996: 8) points out, HPSG has as its main goal to provide illuminating syntactic analysis, rather than providing computational implementations. He also notes that HPSG has to be adapted before it can be used in computer modelling, as is also the case with other frameworks.

2. Formalism

This section provides the reader with an idea of the formalism and implementation of HPSG. It is noted that there is a lot more to HPSG analyses and that this is to be used only as a flavor of the kind of formal entities used in HPSG. The reader is referred to the references cited here and the recommended readings for a more complete view of the formalism.

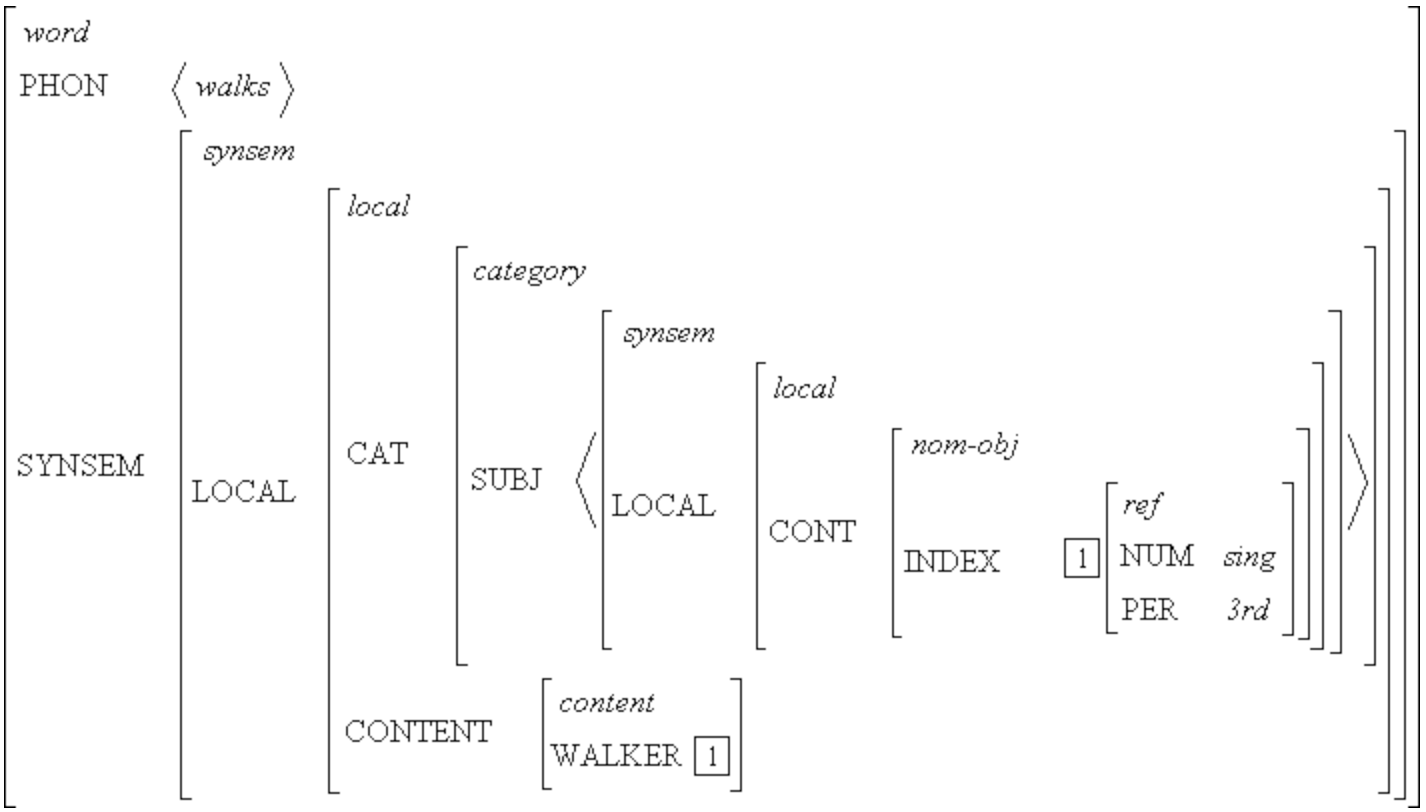
HPSG representations use feature structures, often written as attribute-value-matrixes (AVMs)[4], to represent grammar principles, grammar rules and lexical entries. A constituent is licensed if it is described by a feature structure and this feature structure conforms[5] to each grammatical principle. When the constituent is phrasal, it also has to conform to a grammar rule and when it is lexical, it has to conform to a lexical entry (Cooper, 1996: 192).

2.1. The sign and its features

An important concept in HPSG representations is that of a sign. A sign is a collection of information, including phonological, syntactic and semantic

constraints, which is what is represented in AVMs (Bouma, van Eynde and Flickinger, 2000: 44). AVMs encode feature structures where each attribute (feature) has a type and is paired with a value (Carnie, 2002: 360). The notion of sign is formalized by being the type of every constituent admitted by HPSG (Cooper, 1996: 192), including both words and phrases. Signs receive the subtypes *word* or *phrase* depending on their phrasal status. These subtypes differ in that they conform to different constraints (Ginzburg and Sag, 2000: 2), but both contain attributes such as phonology (PHON) and syntax/semantics (SYNSEM). PHON has as its value a list of phonological descriptions. SYNSEM (our focus here) has another AVM typed *synsem* as its value, which in turn contain other attributes that can have other AVMs as values (Bouma, van Eynde and Flickinger, 2000: 2). (1) illustrates what the AVM of a HPSG sign may look like:

(1)



(Riehemann, 1995: 1)

Notation of AVMs may vary[6]. For example, types may be omitted and the AVMs may be made simpler (or much more complex!) than the one above, depending on what aspects are relevant to each analysis. In (1), attributes (written in capital letters) are followed by their values, which can be atomic (such as *sing*) or complex (such as the values of SYNSEM, INDEX, etc which are represented as AVMs themselves). The types are expressed in lower

case letters on the top part of the brackets (e.g. *word*) (Riehemann, 1995: 2 - 8). Angled brackets (< >) indicate ordered lists.

The boxed number found in the values of INDEX and WALKER is a (*coreferential*) *tag*, used to indicate that certain substructures are identical. The tag in the description of the index attribute indicates that the tagged AVM is identical[7] to the AVM which would be the value for the walker attribute. In other words, the number and person features of the head (verb) must match those of WALKER. Tags perform a function analogous to feature checking in P&P and Functional Control in LFG[8] (Carnie, 2002: 361).

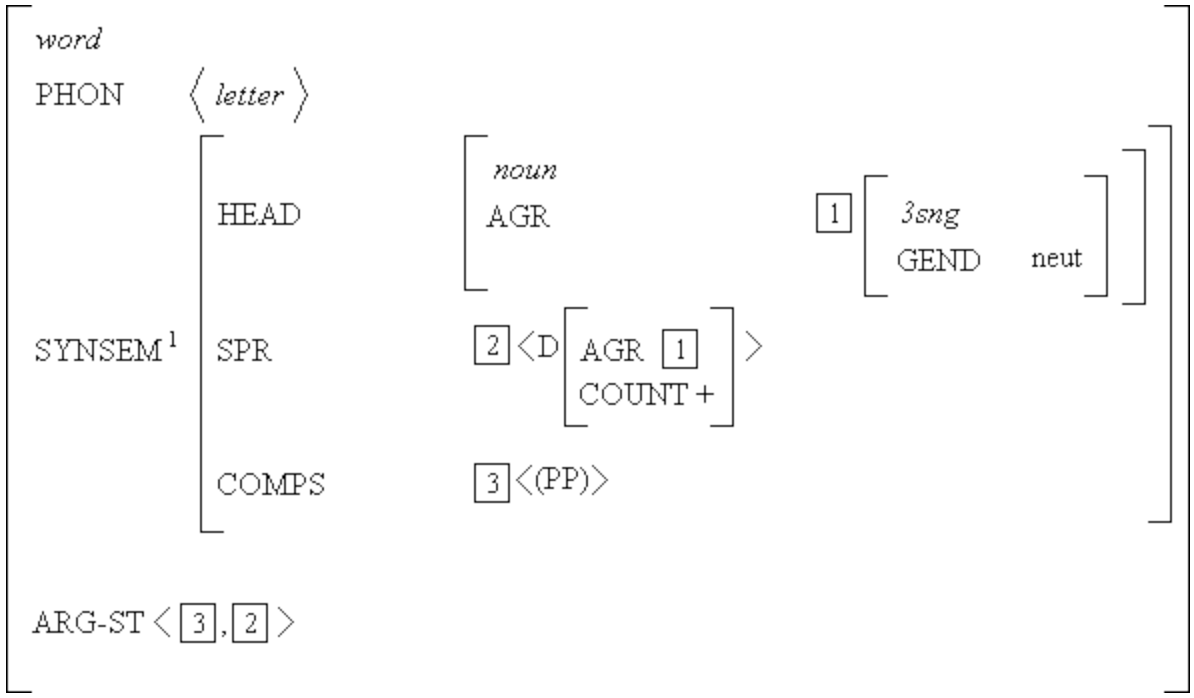
One of the main functions of the SYNSEM attribute is to encode the formal grammatical features of a constituent (Carnie, 2002: 360). Broadly speaking, it can be said that the value of SYNSEM gives the syntactic category of a constituent, which is a complex feature structure[9], as hinted in the AVM in figure 1. As mentioned above, within these feature structures, each attribute often has other feature structures containing more attributes as their values. One of the attributes embedded within SYNSEM is the HEAD feature. This feature encodes all the syntactical features that a head and its phrasal constituent have in common (this interacts with the head principle, discussed below) (Cooper, 1996: 192), including the inflectional properties of a constituent and its part-of-speech. Other examples are the specifier (SPR) and complement (COMP) features, which restrict the elements that may appear as the specifier and as the complement in a constituent. (Carnie, 2002: 361).

Information about specifiers and complements is also[10] present in the argument structure attribute (ARG-ST), which encodes information roughly equivalent to that contained in theta-grids in P&P. Its value is an ordered list of the arguments required by the sign, which may contain specific selectional restrictions (Carnie, 2002: 362). For example, the fact that the inflected form *loves* requires a third person subject can be encoded in the following way:

(2) $\langle loves, [\text{ARG-ST} \langle \text{NP}, \text{NP} \rangle] \rangle$

The following simplified partial AVM illustrates the features discussed in this section:

(3)



(Adapted from Carnie, 2002: 363)

For a more detailed (though still summarized) schematic representation of a sign, click here.

2.2. Rules, principles and unification

Feature structures such as those described in the previous section interact with rules and principles to determine well-formed expressions of a language (Kim, 2000: 8). Principles and rules limit what signs are possible expressions of a language. Principles apply to all signs, whereas grammar rules apply to a subset of signs, such as phrases (as mentioned in the beginning of section 2, words have to conform to lexical entries instead of rules) (Cooper, 1996: 192-193). Although principles and rules can be paraphrased in words, as is done below, they are implemented by feature structures (like those described in section 2.1) which can be compared to specific signs to check whether or not they are well formed. Roughly speaking, this is accomplished by checking on whether or not the AVM of the sign fits with the AVM imposed by the principles and rules (Riehemann, 1995: 3-4).

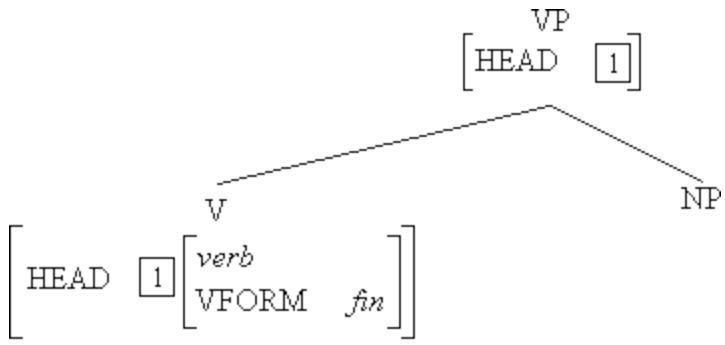
One example of a HPSG principle is the Head Feature Principle. The Head Feature Principle ensures that the properties of heads (such as part-of-speech, case and verb inflection) of a head are projected onto headed phrases (Kim, 2000: 8-9):

(4) Head Feature Principle (HFP):
The HEAD value of a headed phrase is identified with that of its head-daughter.

(Click here to view an AVM representation of the head principle, abbreviated using a path notation.)

This can be illustrated in the following graph, where the tag [1] indicates that the HEAD value of the verb is identical to the HEAD value of the VP:

(5)



(Kim, 2000: 9)

The effects of the Head Principle, combined with the effects of the Valence Principle lead to a result analogous to that obtained by the use of X-bar theory bar levels in P&P. (Click here to read more about the Valence Principle and see other HPSG Principles).

As mentioned in section 1, HPSG is a unification-based formalism. This is because it uses unification as its combinatorial procedure. Unification is a means to combine the information of two AVM descriptions together. In order to unify two AVMs, one simply puts the information of the two AVMs together, resulting in the most general description which satisfies both A and B. That is, the result of this operation is the logical intersection of the features that satisfy the first AVM_1 and the second AVM_2 (AVM_1 & AVM_2). If the information in the two AVMs is incompatible, the intersection will be empty and unification is said to fail. For example, the unification of (6) and (7) gives (8):

(6) [NUM *sing*]

(7) [PER *3rd*]

(8) $\begin{bmatrix} \text{NUM} & \textit{sing} \\ \text{PER} & \textit{3rd} \end{bmatrix}$ (Riehemann, 1995: 5)

However, the unification between (9) and (10) fails because they are incompatible.

(9) $\begin{bmatrix} \text{NUM} & \textit{sing} \end{bmatrix}$

(10) $\begin{bmatrix} \text{NUM} & \textit{plur} \end{bmatrix}$ (Riehemann, 1995: 5)

(For an example of unification in the presence of tagged AVMs, click here).

The checking of principles discussed in this section is done though unification: if the unification between the feature structures described by the principle and a particular AVM fails, then the principle is not satisfied (Riehemann, 1995: 4-5). Thus, unification seems to be an analogue to feature checking in more recent versions of P&P (i.e. Minimalism).

2.3. The lexicon

The lexical entries contain a large amount of information. Lexical entries are fully inflected, as it can be seen from the simplified lexical entries below:

(11) gave
$$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} \\ \text{VFORM} & \textit{fin} \end{bmatrix} \\ \text{SUBJ} & \langle \text{NP}\,[\textit{nom}] \rangle \\ \text{COMPS} & \langle \text{NP}[\textit{acc}], \text{PP}[\textit{to}] \rangle \end{bmatrix}$$

(12) books (Kim, 2000: 14)
$$\begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{SPEC} & \langle \text{DetP} \rangle \\ \text{COMPS} & \langle\,\rangle \end{bmatrix}$$

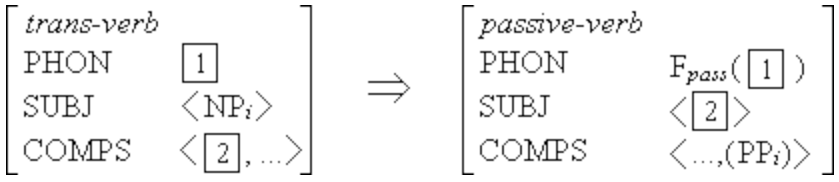
If the lexicon were just an unorganized collection of entries such as the one above, there would be a lot of redundancy and many generalizations would be missed. For example, there would be one entry for the word *book* and another completely unrelated entry for the word *books*. The lexicon would miss the fact that these words are related according to a recurrent pattern (in this case plural inflection). This kind of redundancy is known as *horizontal*

redundancy. Another example would be two lexical entries for the passive and the active form of a verb. Another type of redundancy, known as *vertical redundancy*, is caused by encoding in each lexical entry a lot of linguistic information that is shared with whole world classes (Kim, 2000: 15, citing Pollard and Wasow, 1985, and Pollard and Sag, 1987). An example would be the redundancy of listing the need for a determiner in all singular count nouns. HPSG uses hierarchical classification and lexical rules to deal with redundancy and organize its lexicon.

Hierarchical classification is used to deal with vertical redundancy and lexical rules to deal with horizontal redundancies. Hierarchical classification works by assigning a type (*sort)* to words of specific categories and identifying them with a category which covers a group of words (the category is known as a *supersort*). Constraints are assigned to a supersort (the categories) and inherited by all the sorts associated with it (i.e. all the words of a certain category). Thus, instead of having constraints repeated in each lexical entry, the lexical entry would list its sort and the constraints belonging to that group of words (supersort) would be applied to it.

Lexical rules deal with vertical redundancy by reducing the number of lexical entries necessarily stored. They are used to generate new lexical entries from basic lexical entries (Kim, 2000: 15-17). For example, the passive lexical rule seen in (13) generates a passive verb from a transitive verb:

(13)



(Kim, 2000: 16)

This aim of this section was to give the reader an idea of issues in lexical organization. Again, the reader is referred to the works referred to here and the recommended readings for a more complete discussion.

2.4. A simplified example

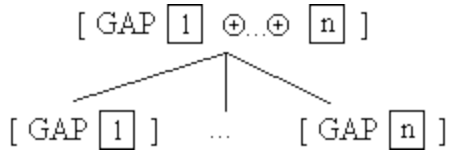
Because of the degree of precision of HPSG, the reader should develop their background further than this introduction before reading original HPSG analyses of specific phenomena. However, here is a very quick description of an account of long distance dependencies as an example (taken from Carnie,

2002: 369 – 371).

P&P deals with the dependencies between the surface position of *wh*-phrases or topicalized constituents and the position with which they would be thematically associated by recurring to movement. Instead of movement (which is not used in this framework), HPSG uses a GAP feature, which indicates that a required argument is missing from a particular structure.

The arguments required by the argument structure (encoded as the value of ARG-ST, seen in section 2.1) appear in the value of COMP (the complement attribute mentioned in section 2.1) in sentences without long distance dependencies. In structures where arguments in ARG-ST are missing, the missing arguments appear in the value of GAP, instead of COMP. Thus, if the value of GAP is not empty, it indicates that there is not an argument in the expected complement position. The GAP principle (14) forces the GAP feature to percolate up to the mother of the node containing a non-empty GAP:

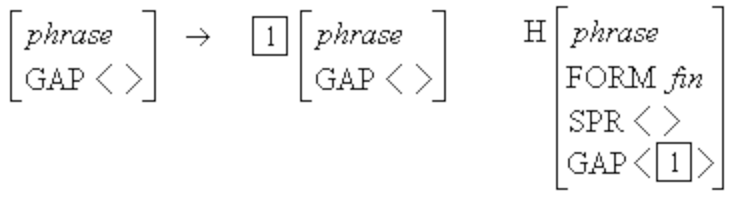
(14)



(Carnie, 2002: 369)

The GAP feature percolates up, but it needs to be satisfied in a well-formed sentence. This can be done by applying the Head Filler Rule:

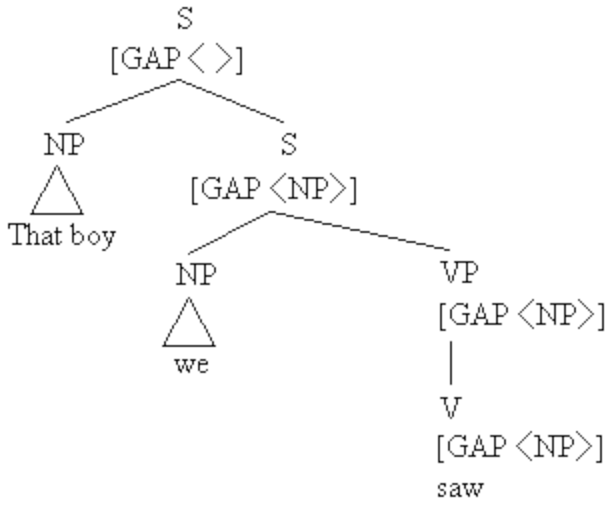
(15)



(Carnie, 2002: 370)

As an example, the topicalized sentence in (16), could be summarized in (17)

(16) That boy we saw

(17) 

(Carnie, 2002: 371)

In (17), the verb 'see' requires a complement which is not present, thus the GAP feature is filled (by the NP required by the argument structure of the verb. This feature is percolated up to the level of the sentence, where the Head Filler Rule is applied, making the value of the gap empty.

3. Similarities and differences between HPSG and P&P

As mentioned in section 1, HPSG and P&P have the same broad goal of building a scientific theory of language. These frameworks are also similar in that P&P concepts often have HPSG analogues and vice versa. Examples of these are P&P theta-grids and HPSG ARG-ST (Pollard, 1997: 1-2), and HPSG unification and the Minimalist idea of feature checking (Carnie, 2002: 365). Borsley (1991: 208) argues that another important similarity between these approaches is that they agree in their views of which linguistic phenomena should be considered a central point of investigation. For example, both approaches investigate similar sentence-types and often agree, in broad terms, on the properties of these sentences (how they are like and how they are unlike other sentences-types)[11]. However, these approaches differ on a number of points.

One difference in the models implemented by both frameworks is that P&P has multiple levels of representation[12] (it is a multistratal framework), whereas HPSG uses only one level (Borsley, 1991: 202 – 206). Another difference is that, unlike in P&P, there is no distinction between terminal and non-terminal nodes in HPSG. This is related to the fact that HPSG is “fractal”, meaning that it is structurally uniform as parts get smaller: every sign down to the word level has syntactic, semantic and phonological features encoded in a

similar way[13] (Pollard, 1997: 5). Other differences are that HPSG avoids the use of movement and the use of null elements, such as complementizers and functional heads, whereas these are widely used in P&P analyses. Borsley (1991: 9) points out that these similarities are often obscured by differences in terminology and notation.

There are a number of other technical differences between these two frameworks. Some of these may be related to (or a consequence of) a more fundamental difference in the how advocates of the two approaches believe the major goal of building a theory of linguistics should be pursued. HPSG takes a bottom-up approach to language investigation, making very precise hypotheses with the aim of making them easily falsifiable. That is, proponents of HPSG start their claims from specifics and tend to be more conservative in their generalizations (Pollard, 1997: 2). They often criticize the P&P approach arguing that it is too vague to be empirically verifiable (e.g. Pollard (1997 and 1996), Borsley (1996)). On the other hand, P&P advocates may criticize those of HPSG in that their focus on formal precision and the conservative nature of their claims comes at the cost of loss of the explanatory value of their analyses (that is HPSG analyses are more precise in their description but not as interesting in offering insights on the reasons why languages are the way they are) (Horrocks, 1987: 298-299).

However, the conflicting views of these approaches may not be completely incompatible. The different immediate goals of these theories can be seen as two approaches tackling the same ultimate goal from different angles. Horrocks (1987: 299) argues that the focus of HPSG in descriptive adequacy and the P&P focus in explanatory adequacy are complementary (and neither is necessarily better than the other in absolute terms). Although these approaches differ in their goals and in many technical aspects of their formalism, it seem that differences are often exaggerated (Borsley, 1991: 208).

This section provided a brief comparison between P&P and HPSG. The reader will encounter more differences when reading specific HPSG analyses. Click here for a short overall comparison between HPSG, P&P and LFG. For interesting discussion on the differences between these frameworks and more details on the information provided in this summary, consult the references.

References

Borsley, Robert D. 1991. *Syntactic Theory: A Unified Approach*, London:

Edward Arnold.

Borsley, Robert D. 1996. *Modern Phrase Structure Grammar*, Cambridge, MA: Blackwell Publishers Ltd.

Borsley, Robert D. (ed.). 2000. *The Nature and Function of Syntactic Categories,* Syntax and Semantics 32. New York: Academic Press.

Bouma, Gosse, Frank van Eynde and Dan Flickinger. 2000. Constraint-Based Lexica. In F. van Eynde and Dafydd Gibbon (eds.), *Lexicon Development for Speech and Language Processing*, 43-75. Dordrecht: Kluwer.

Carnie, Andrew. 2002. *Syntax: A Generative Introduction*, Oxford: Blackwell Publishing.

Cooper, Richard P. 1996. Head-Driven Phrase Structure Grammar. In K. Brown and J. Miller (ed.), *Concise Encyclopedia of Syntactic Theories*, 191-196. Oxford: Pergamon.

Ginzburg, J. and Ivan A. Sag. 2000. *Interrogative Investigations: The Form, Meaning and Use of English Interrogative Constructions*. Chapter 2: HPSG – Background. Stanford: CSLI Publications. <http://www-csli.stanford.edu/~sag/L221a/gs-ch2.pdf>

Haegeman, Liliane. 1994. *Introduction to Government and Binding Theory*, Oxford: Blackwell Publishing.

Horrocks, Geoffrey. 1987. *Generative Grammar*, Essex: Longman.

Kim, Jong-Bok. 2000. *The Grammar of Negation: A Constraint-Based Approach*. Chapter 1: Introduction and Theoretical Foundations. Stanford: CSLI Publications.

Marantz, Alec. 1995. The Minimalist Program. In Gert Webelhuth (ed.), *Government and Binding Theory and the Minimalist Program*, 349-382. Cambridge, MA: Blackwell Publishers.

Pollard, Carl. 1996. *The Nature of Constraint-Based Grammar*. Unpublished manuscript: Ohio State University. <http://www-csli.stanford.edu/~sag/L221a/pollard-96.txt>

Pollard, Carl. 1997. *Lectures on the Foundations of HPSG*. Unpublished manuscript: Ohio State University. <http://www-

csli.stanford.edu/~sag/L221a/cp-lec-notes.pdf>

Riehemann, Susanne. 1995. *The HPSG Formalism*. Unpublished manuscript: Stanford University. <http://www-csli.stanford.edu/~sag/L221a/hand2-formal.pdf>

Sells, Peter. 1985. *Lectures on Contemporary Syntactic Theories*, Stanford: CSLI Publications.

Shieber, Stuart M. 1986. *An Introduction to Unification-Based Approaches to Grammar*, Stanford: CSLI Publications.

Top of the Page
HPSG Front Page

---

[1] This overview presents a very simplified version of HPSG and some of the work on which it is based is also simplified in order to make it accessible to beginning students (e.g. Carnie, 2002: 359 – 375). In other words, this summary can be used to give the reader an idea of some basic concepts of HPSG, which will hopefully be helpful before pursuing a more detailed understanding from other work.

[2] In this website, the abbreviation P&P (Principles and Parameters) refers to various versions of Chomskyan approaches, such as GB (e.g., Haegeman, 1993) and Minimalism (e.g. Marantz, 1995). It is not claimed that these are equivalent, but a discussion of the differences between them is outside the scope of this work. This group will be compared to HPSG and LFG as a whole.

[3] HPSG is declarative, meaning that it provides a model of what linguistic entities are possible, rather than how they are processed (Shieber, 1986: 7), which distances it from claims based on the psychology of language processing (Cooper, 1996: 191). This is also true of P&P approaches (Pollard, 1997: 1).

[4] Although AVMs are widely used in the HPSG literature, it is important to note that they are an abbreviation of a more precise feature logic (Pollard, 1997: 4).

[5] Conformity is checked through unification (Shieber, 1986).

[6] Refer to Riehemann (1995) for a more mathematical discussion of AVM notation.

[7] It is important to note that the information sharing indicated by these tags is not the same as repeating the tagged information twice. Simply repeating the information has different computational consequences (Riehemann, 1995: 3-4) - for a simple example click here.

[8] But they have different implications. Carnie (2002: 361) points out that the HPSG allows for a non-transformational analysis of raising and the elimination of PRO in control constructions.

[9] For an accessible discussion of HPSG categories, the reader is directed to Borsley (1996: 24-40). Borsley presents the advantages of complex over atomic categories and compares HPSG

categories with GPSG and P&P categories. For a deeper discussion on categories across different frameworks, including HPSG, the reader is referred to a volume edited by Borsley (2000).

[10] Carnie (2002: 362) notes that this apparent redundancy is necessary for accounts of binding and that Manning and Sag (1998) discuss other reasons to differentiate ARG-ST from COMP and SPR.

[11] Borsley (1991: 208) cites the constructions in his whole book as examples of these construction. These include sentences with control, raising, *wh*-dependencies, island constraints and others. For a thorough comparison of HPSG and P&P views on specific phenomena refer to that volume (Borsley, 1991).

[12] This may have been more pronounced in earlier versions of Chomskyan approaches which assumed an additional level of a Deep-Structure which was the mapped into a Surface-Structure. However, the distinction still applies to recent Minimalist developments: even though there is no distinction between Surface-Structure and Deep- Structure, there is still a point (Spell-out) in which the derivation splits into two levels, PF and LF. Furthermore, structures are still derived from other structures by operations such as move-a, even before Spell-out. For more details on levels of representation in Minimalism, refer to Marantz, 1995).

[13] Pollard points out further that this leads to the lack of the issue of late versus early lexical insertion, recently brought up in Minimalist approaches, in HPSG.