HPSG and Natural Language Processing

Introduction

Head-Driven Phrase Structure Grammar (HPSG) is the grammatical theory developed by Carl Pollard and Ivan Sag during the mid1980s [1, 2]. It was developed by synthesizing several contemporary linguistic theories, including Categorial Grammar (CG), Generalized Phrase Structure Grammar (GPSG) [3], and Lexical Functional Grammar (LFG) [4], from which it borrows some interesting ideas. The HPSG architecture has been largely pursued by an increasing number of linguists, since the formally well-defined framework allows for an explicit formalization of a linguistic theory.

HPSG, an integrated theory of natural language syntax and semantics, is a feature-based grammatical framework characterized by a modular specification of linguistic generalizations through extensive use of principles and a grammatical information lexicon.

HPSG is formulated in terms of order-independent constraints. In other words, the grammar is formulated as a declarative system of constraints. These constraints provide partial grammatical information that can be flexible, and consulted in a variety of language processing models based on the notion of incremental, on-line integration of heterogeneous types of information. Key advantages of HPSG appropriate for use in natural language processing are as follows:

- *Small number of rules and rich-information lexicon*
  Based on the assumption of a universal syntax, a small number of highly schematic syntactic rules are assumed to apply universally. The task of explaining variations between languages must be carried out in the lexicon [5]. The detailed lexical entries of HPSG are concisely expressed within a multiple inheritance hierarchy and lexical rules. Such hierarchical lexicons allow cross-cutting generalizations about words to be expressed in a highly efficient and compact organization.
- *Language-independent principles*
  The modular design of HPSG offers a large degree of flexibility for applying the framework to new languages and changing individual components of a grammar.
- *Unification-based constraints*
  Work in the unification framework [6] has shown a formalism based on unification is particularly well-suited for declarative modes of problem solving. In addition, a declarative mode allows neutral grammar constructions in processes applying to them, whether parser or generator [7].
- *Local encoding of unbounded dependencies*
  Filler-gap phenomena and other long-distance dependencies are treated in terms of certain feature specifications present throughout the `path' from filler to gap [8].

1. Words and Phrases as Feature Structures

Utterances in HPSG are modeled in terms of feature structures of type *sign*, with its two immediate subtypes *word* and *phrase* [9]. The lexical entries are descriptions of feature structures of type *word* while phrase structure rules are partial descriptions of feature structures of type *phrase*.

A feature structure is a description of an object. It specifies some or all of the information asserted to be true of the object. The features CATEGORY (CAT) and CONTENT (CONT) (Figure 1) specify the syntactic and semantic information of an object respectively. Figure 1 represents a partial description of the lexical entry *hide* in an attribute-value matrix (AVM) diagram.[1] Each feature takes a value of a particular and appropriate type. The grammar must include a specification of what types are included, which features are appropriate for which type, and what type of value is appropriate for each feature. Some types, their features, and their value types are shown in Table 1.

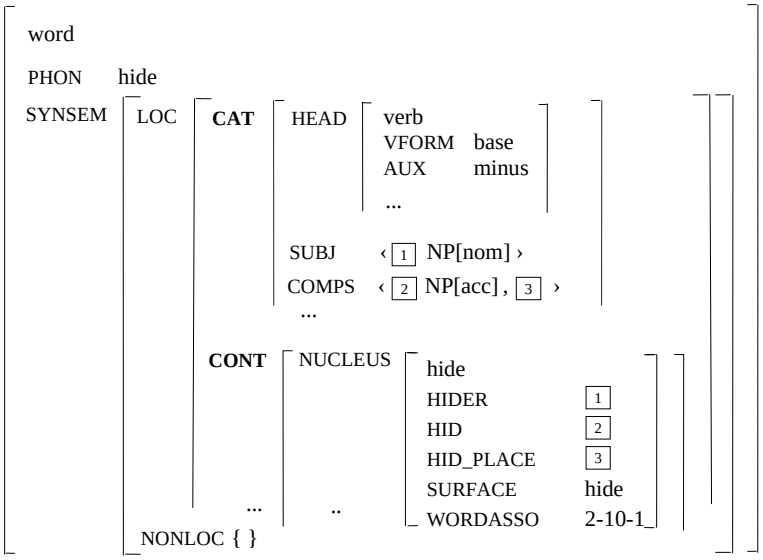

Figure 1. The lexical entry *hide*

Figure 2 presents a (simplified) partial description of the phrase *Leslie drinks milk* [9] in terms of typed feature structures.

This representation indicates *Leslie drinks milk* is the type of head-subject phrase[2] (hd-subj-ph). The hd-subj-ph is a subtype of the type headed-phrase (hd-ph). The hd-ph is the type of (hd-comp-ph), *head-adjunct-phrase* (hd-adjunct-ph). Instances of

[1] The AVM diagram is the standard method of representing grammatical information in modern computation grammar theories.

[2] There are two types of phrases in English: headed phrase (e.g., head- subject phrase, head-complement-phrase, head-modifier-phrase) and nonheaded phrase (e.g., imperative-phrase, coordinate-phrase). Further details can be found in Ginzburg and Sag 1998 and Sag and Wasow 1999.

Table 1. Some types, their features and their value types

Type	Features/Type of value	Immediate Supertype
sign	PHON list(speech-sound) SYNSEM synsem	entity
phrase	...	sign
word	...	sign
synsem	LOC loc NONLOCAL nonlocal	mod_synsem
loc	CAT cat CONT cont CONX conx	entity
cat	HEAD head SPR list_synsem SUBJ list_synsem ...	entity
hd-ph	HD-DTR sign NON-HD-DTRS list(sign)	phrase

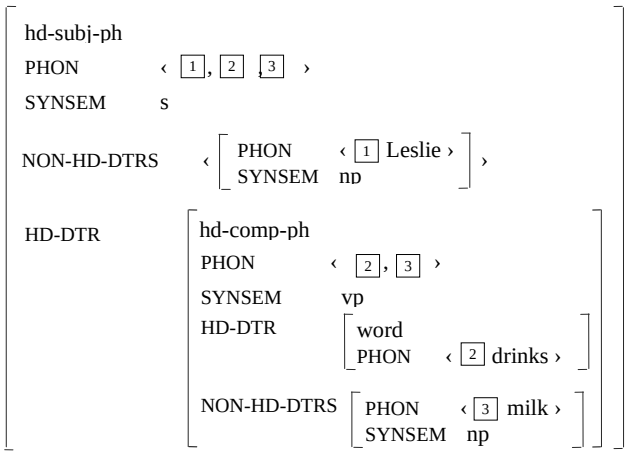

Figure 2. The phrase *Leslie drinks milk*

the hd-ph are governed by the feature declarations shown in the last row of This representation indicates *Leslie drinks milk* is the type of head-subject phrase (hd-subj-ph). The hd-subj-ph is a subtype of the type headed-phrase (hd-ph). The hd-ph is the type of (hd-comp-ph), *head-adjunct-phrase* (hd-adjunct-ph). Instances of Table 1. The indices , , specified in the feature PHONOLOGY (PHON) called tags indicate *structure sharing* between feature values: two or more different features within the feature structure may have their values specified by one and the same feature

structure. The effect of structure sharing is to force two (or more) distinct nodes in a tree admitted by a rule to have identical values for a given feature. In Figure 2, the first phonology (phonology here means phonological shape) of the hd-subj-ph is the same as the phonology of its NON-HD-DTRS which, in this case, is *Leslie*. The second and the third phonologies are the same as those of its HD-DTR which are *drinks* and *milk* respectively. The syntactic category of *Leslie drinks milk* is a sentence (synsem: s). The syntactic category of the NON-HD-DTRS (Leslie) of the given phrase is a noun phrase (NON-HD-DTRS:SYNSEM: np). The syntactic category of its HD-DTR (drinks milk) is a verb phrase (HD-DTR:SYNSEM: vp). The type of this verb phrase *drinks milk* is an hd-comp-ph with the HD-DTR, *drinks*, and the NON-HD-DTRS, *milk*.

2. Lexicon

The lexical types e.g., noun, verb, adj, are a type inheritance. The lexical rules allow complex lexical information (as shown in Figure 1) to be derived via the logic of the lexicon, rather than simply stipulated.

2.1 Lexical Type

The lexical types were introduced to define feature appropriateness, to avoid having to specify values for features that are irrelevant to particular classes.
Lexical *Rule*
The *lexical rule* is a mechanism for further reducing redundancy and stipulation in the
Table 2 illustrates the type constraints that state general properties (in terms of particular feature-value specifications) of particular lexemic types, noun and verb. Noun and verb are subtypes of substantive[3] (subst). The feature CASE is appropriate only for nouns (in English) and its value is either *nominative* (nom) or *accusative* (acc). The features VERB-FORM (VFORM), INVERTED (INV) and AUXILIARY (AUX) are specifiable only for verbs. The forms of verbs can be classified as base, finite, gerund, infinite, passive participle, present participle, and past participle form [Pollard and Sag 1987]. The feature INV and AUX are used to distinguish auxiliary (helping) verbs from all others. The values of INV and AUX are the type of *boolean*. In interrogatives, a finite auxiliary verb precedes the subject, therefore, the INV and AUX values of a finite auxiliary verb are *plus* while the INV and AUX values of other kinds of verb are *minus*.

Each type in the lexical hierarchy has constraints associated with it. Some inviolable, and others default in nature. The inheritance of constraints in this type hierarchy is *default*. The *default inheritance* allows contradictory information associated with a subtype to take precedence over (or override) constraints that would otherwise be inherited from a supertype. In other words, lexical items have many properties in common. They may differ from one another in terms of particular constraints that override the general constraints governing their supertypes. By organizing the lexicon in terms of a type hierarchy and the use of *default inheritance* of

[3] According to Pollard and Sag, noun, verb, adjective, preposition, relativizer are subtypes of *substantive*, whereas marker (e.g., complementizers) and determiner are subtypes of *functional* [Pollard and Sag 1994].

constraints, the stipulations associated with particular lexical entries can be minimized and the shared properties of different word classes can be expressed.

2.2 Lexical Rule

The *lexical rule* is a mechanism for further reducing redundancy and stipulation in the

Table 2. Features and their value types of the types noun and verb

Type	Features/Type of value	Immediate Supertype
noun	[CASE case]	subst
verb	[VFORM vform INV boolean AUX boolean]	subst

lexicon by using information in one lexical entry as the basis for generating another lexical entry. Lexical rules are used for deriving predictably related lexical entries, e.g., inflected forms of verbs and nouns. A lexical rule applies to a lexical entry (of type *word*) and produces as output a new lexical entry whose (morphological) form, syntactic category and semantics are systematically related to the input [10]. Figure 3 illustrates the (simplified) rule that applies to verb bases in English, giving their 3rd-singular verb form. This 3rd-singular verb lexical rule is taken from hpsg.pl written by Gerald Penn [11]. The rule suffixes an -s or -ies (in case the verb ended with y), thus it generates *hides* for *hide* and *flies* for *fly*. The rule says that for every verb whose form is uninflected (VFORM: bse), not an auxiliary verb (AUX: minus) and it takes a nominative as its subject (SUBJ: NP (nom)), there is a corresponding lexical entry for a 3rd-singular verb. The form is dictated by the 3rd-singular verb lexical rule: finite (VFORM: fin) with a nominative and singular subject (SUBJ: NP (nom, sg)).
Figure 4 illustrates *hides*, the output of the 3rd-singular verb lexical rule applied to the lexical entry *hide* shown in Figure 1.

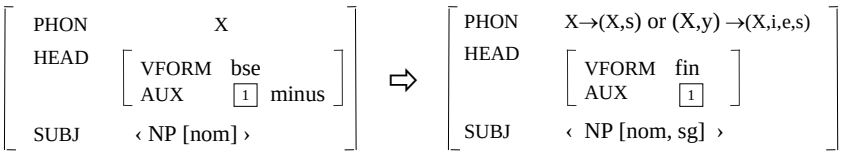

Figure 3. The 3rd-singular verb lexical rule

3. Universal Principles

The Head Feature and Semantics principle are described in this section.

3.1 Head Feature Principle

Head Feature Principle (HFP) can be formulated as a constraint on phrases of the type hd-ph. This principle restricts sharing the HEAD feature between a mother sign and its

head daughter as shown in Figure 5. HEAD always takes as its value a feature structure appropriate to each category as discussed in the last section (see the values of the HEAD features of *drinks* and *milk* in Figure 6). The effect of the sharing restriction is to guarantee headed phrases really are a *projection* of their head daughters. In Figure 6, the HD-DTR, *drinks*, of the hd-comp-ph, *drinks milk*, passes its HEAD feature's value to the HEAD feature of its mother (marked by the dotted ellipse). The HD-DTR, *drinks milk*, of the hd-subj-ph, *Leslie drinks milk*, passes its HEAD feature's value to the HEAD feature of its mother. HFP also indicates the

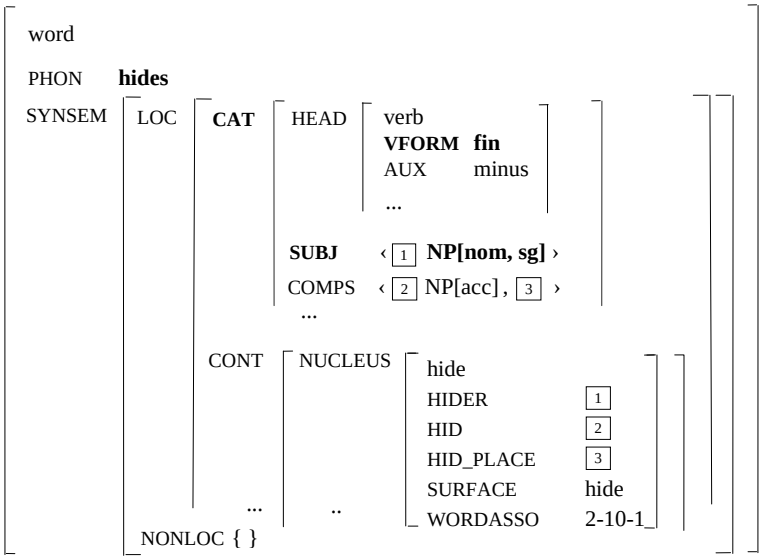

Figure 4. Example output, *hides*, of the 3rd-singular verb lexical rule

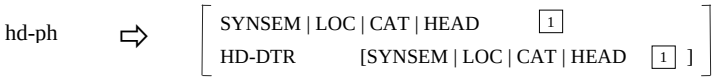

Figure 5. Head Feature Principle

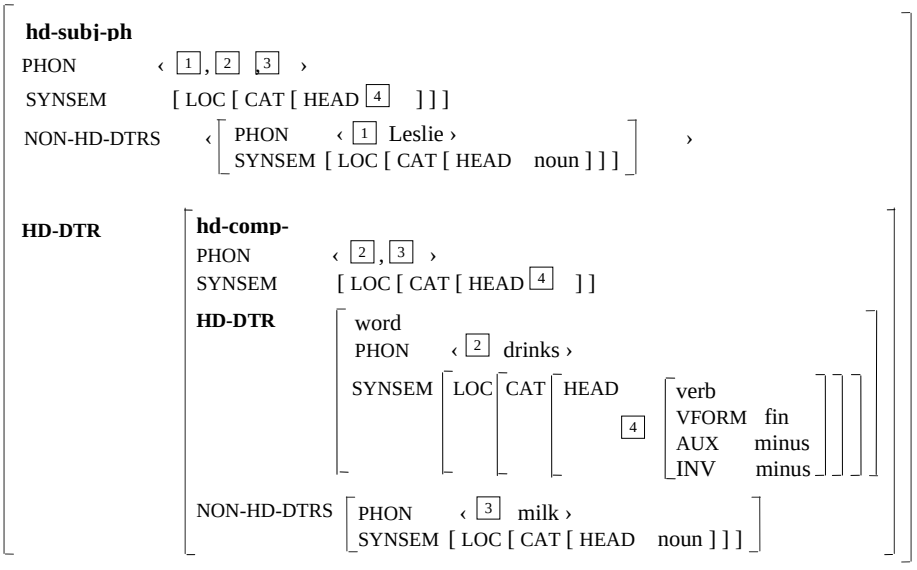

Figure 6. HFP applied to *Leslie drinks milk*

category of verb phrases are verb because they have verbal heads. Noun phrases are nominal because they have noun heads.

3.2 Semantics Principle

Pollard and Sag guarantee the semantics of a mother sign constrained by the feature CONT are identified with the adjunct daughter (ADJUNCT-DTR) if the phrase is the type of head-adjunct-phrase (hd-adjunct-ph) [2]. However, if the phrase is not the type of hd-adjunct-ph then the semantics of a mother sign are identified with the HD-DTR (2). The features CONT are concerned principally with linguistic information that bears directly on semantic interpretation. The CONT value of nominals (e.g., lexical nouns and their phrasal projections) is the feature structure of INDEX and restriction (RESTR). For more principles e.g., Spec Principle, Marking Principle, Nonlocal Feature Principle and Relative Uniqueness Principle, consult [1], [2].

$$\text{hd-ph} \Rightarrow \begin{bmatrix} \text{SYNSEM} \mid \text{LOC} \mid \text{CONT} & \boxed{1} \\ \text{HD-DTR} & [\text{SYNSEM} \mid \text{LOC} \mid \text{CONT} \ \boxed{1}\ ] \end{bmatrix}$$

with the exception of the hd-adjunct-ph

$$\text{hd-ph} \Rightarrow \begin{bmatrix} \text{SYNSEM} \mid \text{LOC} \mid \text{CONT} & \boxed{1} \\ \text{ADJUNCT-DTR} & [\text{SYNSEM} \mid \text{LOC} \mid \text{CONT} \ \boxed{1}\ ] \end{bmatrix}$$

Figure 7. Semantics Principle

References

[1] Pollard, C. and Sag, I. A. (1987) Information-Based Syntax and Semantics, Lecture Notes No. 13, Stanford, Calif: CSLI Publication.

[2] Pollard, C. and I.A. Sag (1994) Head-Driven Phrase Structure Grammar, Center for the Study of language and Information, Stanford, The university of Chicago Press, Chicago & London.

[3] Gazdar, G., E. Klein, G. K. Pullum and I. V. Sag (1985) Generalized Phrase Structure Grammar, Oxford: Basil Blackwell, Cambridge, Maasachusette, Harvard University Press.

[4] Brensnan, J., and Ronald M. Kaplan (1982) Introduction *In* Brensnan, ed.

[5] Carpenter, B. (1991) The Generative Power of Categorial Grammars and head-Driven Phrase Structure Grammars with Lexical Rules, *In* Computational Linguistics, Association for Computational Linguistics, 17(3), p. 301-313.

[6] Shieber, S. M. (1986) An Introduction to Unification-Based Approachs to Grammar, CSLI Lecture Notes Number 4, Center for the Study of Language and Information CSLI, Leland Stanford Junior University.

[7] Estival, D. (1994) Reversible Grammars and Their Application in Machine Translation *In* Reversible Grammar in Natural Language Processing, Kluwer Academic Publishers, The Netherlands, p. 293- 320.

[8] CSLI, (2012), Head-Driven Phrase Structure Grammar, http://hpsg.stanford.edu/ideas.html as of September 2012.
[9] Ginzburg, J. and I. V. Sag, (1998) English Interrogative Constructures (Draft of July) *In* Construction : an HPSG Perspektive by I. A. Sag and A. Kathol, Language Advanced Course, 10th European Summer School in Logic, Language and Information, Saarbrücken, 17- 28 August.
[10] Sag, I. A. and T. Wasow. 1999 Syntactic Theory: A Formal Introduction, CSLI Lecture Notes Number 92, Center for the Study of language and Information, Stanford, California.
[11] Penn, G. (1993) hpsg.pl available at "www.cs.toronto.edu/~gpenn/ale/files/.../hpsg.pl" as of November 2012.