

Grammatiktheorie

Head-Driven Phrase Structure Grammar

Vorlesung 2 (06.11.2001)

Valia Kordoni

Email: kordoni@coli.uni-sb.de

WWW: <http://www.coli.uni-sb.de/~kordoni/courses/ws01-02/>

History of HPSG

Head-Driven Phrase Structure Grammar was developed in the mid-1980s by Carl Pollard and Ivan Sag at Stanford University

1. HPSG1: Pollard and Sag (1987)
Formalism (typed feature structures), subcategorization, LP rules, the hierarchical lexicon
2. HPSG2: Pollard and Sag (1994) Chapters 1-8
The structure of the sign, Control Theory, Binding Theory
3. HPSG2: Pollard and Sag (1994) Chapter 9 “Reflections and Revisions”
Valence features SUBJ, COMPS, SPR
4. Further developments in HPSG:
Unbounded Dependency Constructions (Sag, 1997; Bouma, Malouf, and Sag, 1998), Linking Theory (Wechsler, 1995; Davis, 1997), semantic representation (Copestake, Flickinger, and Sag, 1997), argument realization (Sag and Miller, to appear)

HPSG and its influences

Head-Driven Phrase Structure Grammar has been influenced by contemporary theories:

- Syntax
 1. Generalized Phrase Structure Grammar (GPSG; Gazdar, Klein, Pullum, and Sag, 1985)
 2. Categorical Grammar (CG; Mc Gee Wood, 1993)
 3. Lexical Functional Grammar (LFG; Kaplan and Bresnan, 1982)
 4. Construction Grammar (Goldberg, 1995)
 5. Government and Binding Theory (GB; Haegeman, 1994)
- Semantics
 1. Situation Semantics (Barwise and Perry, 1983)
 2. Discourse Representation Theory (DRT; Kamp and Reule, 1993)

Key properties of HPSG: HPSG vs. “Classical” Phrase Structure Grammar

- Similarities:
 1. Both are **monostratal**.
Every sentence is associated with a single tree structure.
 2. Grammar rules have **local** scope only.
Grammatical statements can only refer to a local tree (one node and its daughters). A tree is well-formed if and only if all its local trees are (this is true of *context-free* PSG only).
- Differences:
 1. HPSG uses **complex categories** while Classical PSG uses simple/atomic ones.
 2. HPSG uses **Immediate Dominance (ID) schemata** and **Linear Precedence (LP)** rules instead of Classical PS rules.
ID rules specify the mother and daughters in a local tree without specifying the order of the daughters. LP rules determine the relative order of the daughters in a local tree without making reference to the mother node.
In addition, HPSG proposes several universal principles to further constrain the set of local trees admitted by the ID schemata.
 3. HPSG analyses include semantic representations in addition to syntactic representations.

Key properties of HPSG: HPSG vs. Transformational Grammar

Transformational Grammar: “Chomskyan” frameworks, most recently formalized as Government and Binding Theory, Principles and Parameters, and Minimalism Programme.

- Similarities:

1. Both try to account for a similar range of data.

Some analyses in HPSG – e.g., Binding Theory – are heavily influenced by earlier proposals in TG.

2. Both are theories of **generative grammar**.

Language is seen as the product of a system of rules and principles rather than a collection of strings to be described. The aim of both theories is to formulate these general principles. Furthermore, the generalizations are meant to say something about human linguistic knowledge.

- Differences:

1. HPSG is **non-derivational**, TG is derivational.

TG analyses start with a base generated tree, which is then subject to a variety of transformations – e.g., movement, deletion, re-analysis – that produce the desired surface structure. HPSG analyses generate only the surface tree. Rule ordering is impossible in HPSG because there is no notion of sequential derivation.

2. HPSG constraints are **local**, TG allows non-local statements.

3. **Complex categories** in HPSG are more complex than in TG.

TG uses atomic categories carrying binary feature specifications. HPSG categories are very elaborated in comparison.

4. HPSG is more committed to precise **formalization** than TG.

5. HPSG is better suited to **computational implementation** than TG.

Key properties of HPSG and their consequences

1. HPSG is monostratal, declarative, non-derivational.
No transformations, no rule ordering. In addition, analyses are surface-oriented, with a desire to avoid abstract structures such as traces and functional categories.
2. HPSG is constraint-based.
A structure is well-formed if and only if it satisfies all relevant constraints. Constraints are not violable, as in Optimality Theory, for example. Furthermore, constraints apply only to local trees, although local constraints can interact to have non-local effects.
3. HPSG is a lexicalist theory.
Strong lexicalism (Scalise, 1984). Word-internal structure (e.g., morphology) and phrase structure are handled separately. Phrasal rules cannot manipulate sub-parts of words. Lexically-governed processes such as valence alternation must be analyzed lexically.
4. HPSG is "head-driven" (more on this later).
5. HPSG is a "unification-based" framework where all linguistic objects are represented as "typed feature structures".

Psycholinguistic evidence

HPSG aims to model our knowledge of language. Support for the model proposed by HPSG comes from the fact that it accommodates several empirical facts about human language processing:

1. Human language processing is **incremental**:
Partial interpretations can be generated for partial utterances.
HPSG constraints can apply to partial structures as well as complete trees.
2. HLP is **integrative**:
Linguistic interpretations depend on a large amount of non-linguistic information (e.g., world knowledge).
The signs used in HPSG (typed feature structures) can incorporate both linguistic and non-linguistic information using the same formal representation.
3. HLP is **order-independent**:
There is no fixed sequence in which pieces of information are consulted and incorporated into linguistic interpretation.
HPSG is a declarative model, so information can be added in any order.

4. HLP is **reversible**:

We can produce and understand the same kinds of utterances.

The grammar of HPSG is process-neutral. It can be used for either production or comprehension.