

Head-driven Phrase Structure Grammar – I

Grammatikformalismen (SS 2013)

Yi Zhang

Department of Computational Linguistics
Saarland University

June 18th, 2013

History of HPSG and its influences

- HPSG1: Pollard and Sag (1987)
Formalism (typed feature structures), subcategorization, LP rules, hierarchical lexicon
- HPSG2: Pollard and Sag (1994) Chapter 1-8
The structure of signs, control theory, binding theory
- HPSG3: Pollard and Sag (1994) Chapter 9 “Reflections and Revisions”
Valence features SUBJ, COMPS, SPR
- HPSG4, HPSG5, ...
Unbounded dependency constructions, linking theory, semantic representation, argument realization, ...

History of HPSG and its influences (cont.)

The development of HPSG is influenced by contemporary theories:

- Syntax

- Generalized Phrase Structure Grammar (Gazdar, Klein, Pullum & Sag, 1985)
- Categorical Grammar (McGee Wood, 1993)
- Lexical-Functional Grammar (Kaplan & Bresnan, 1982)
- Construction Grammar (Goldberg, 1995)
- Government-Binding Theory (Haegeman, 1994)

- Semantics

- Situation Semantics (Barwise & Perry, 1983)
- Discourse Representation Theory (Kamp & Reyle, 1993)x

Similarities

- Both are **monostratal**: every analysis is represented by a single structure
- Grammar rules have **local** scope: mother phrase and its immediate daughters

HPSG vs. “Classical” Phrase Structure Grammar

Differences

- HPSG uses complex categories while classical PSG uses simple/atomic ones
- HPSG specifies **Immediate Dominance (ID)** and **Linear Precedence (LP)** separately
 - ID specifies the mother and daughters in a local tree without specifying the order of the daughters
 - LP determines the relative order of the daughters in a local tree without making reference to the mother
 - Further universal **principles** are specified in HPSG to constrain the set of local trees admitted by the ID schemata
- HPSG analyses include semantic representations in addition to syntactic representations

Similarities

- Both try to account for a similar range of data (e.g. in the development of the Binding Theory)
- Both are theories of **generative grammar**

HPSG vs. Transformational Grammar

Differences

- HPSG is **non-derivational**, TG is derivational
 - TG analyses start with a base generated tree, which is then subject to a variety of transformation (e.g., movement, deletion, reanalysis) that produce the desired surface structure
 - HPSG analyses generate only the surface structure, rule ordering is irrelevant
- HPSG constraints are **local**, TG allows non-local statements
- HPSG uses more complex categories than TG
- HPSG is more committed to precise formalization than TG
- HPSG is better suited to computational implementation than TG

Key Properties of HPSG and their consequences

- HPSG is monostratal, declarative, non-derivational
No transformations, no rule ordering. Analyses are surface oriented, with a desire to avoid abstract structure such as traces and functional categories
- 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)
- HPSG is a lexicalist theory
Strong lexicalism; Word-internal structures and phrase structure are handled separately
- HPSG is a unification-based linguistic framework where all linguistic objects are represented as “typed feature structures”

Psycholinguistic Evidence

- 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
- HLP is **integrative**:
Linguistic interpretations depend on a large amount of non-linguistic information (e.g. world knowledge)
The signs in HPSG can incorporate both linguistic and non-linguistic information using the same formal representation

Psycholinguistic Evidence

- HLP is **order-independent**:

There is no fixed sequence in which pieces of information are consulted and incorporated into a linguistic interpretation

HPSG is a declarative and non-derivational model

- HLP is **reversible**:

Utterances can be understood and generated

HPSG is process-neutral, and can be applied for either production or comprehension

Sign is the basic sort/type in HPSG used to describe lexical items (of type *word*) and phrases (of type *phrase*). All signs carry the following two features:

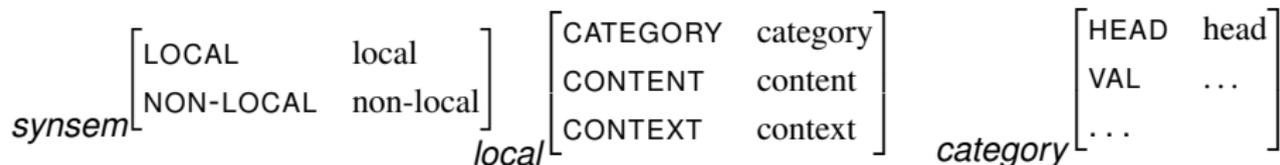
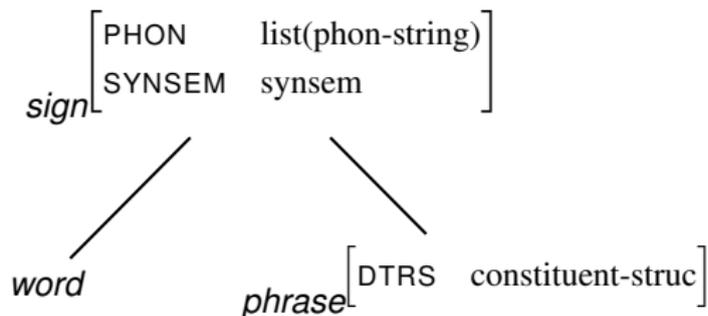
- PHON encodes the phonological representation of the sign
- SYNSEM syntax and semantics

$$\textit{sign} \left[\begin{array}{ll} \text{PHON} & \text{list(phon-string)} \\ \text{SYNSEM} & \text{synsem} \end{array} \right]$$

Structure of the Signs in HPSG

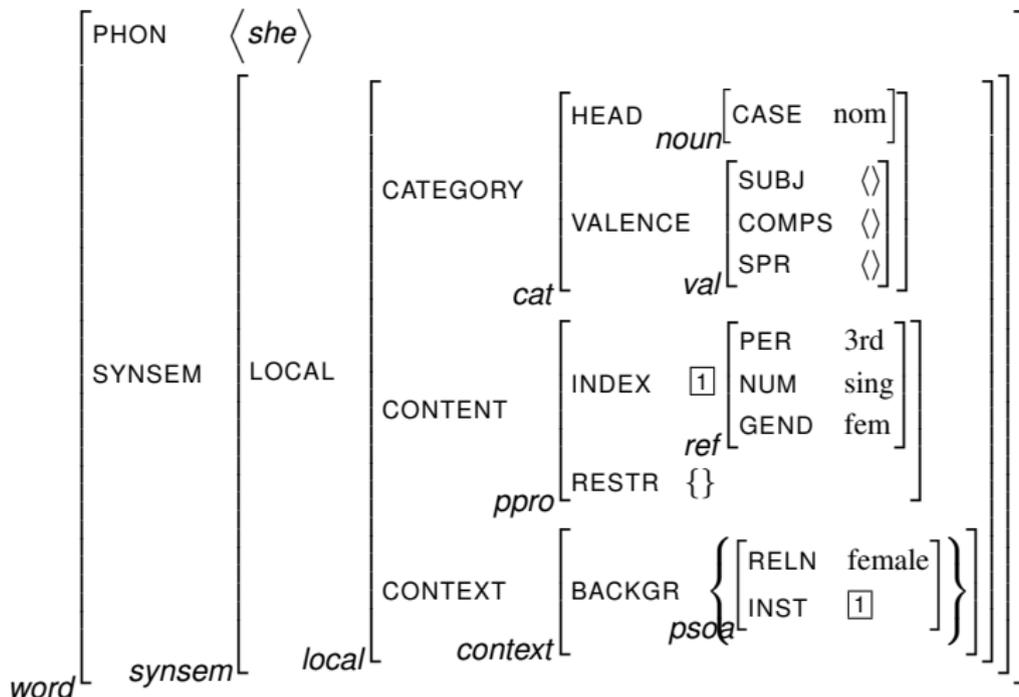
- *synsem* introduces the features LOCAL and NON-LOCAL
- *local* introduces CATEGORY (CAT) CONTENT (CONT) and CONTEXT (CONX)
- *non-local* will be discussed in connection with unbounded dependencies
- *category* includes the syntactic category and the grammatical argument of the word/phrase

An Ontology of Linguistic Objects



Structure of the Signs in HPSG (cont.)

Example



Syntactic Category & Valence

The value of CATEGORY encode information about

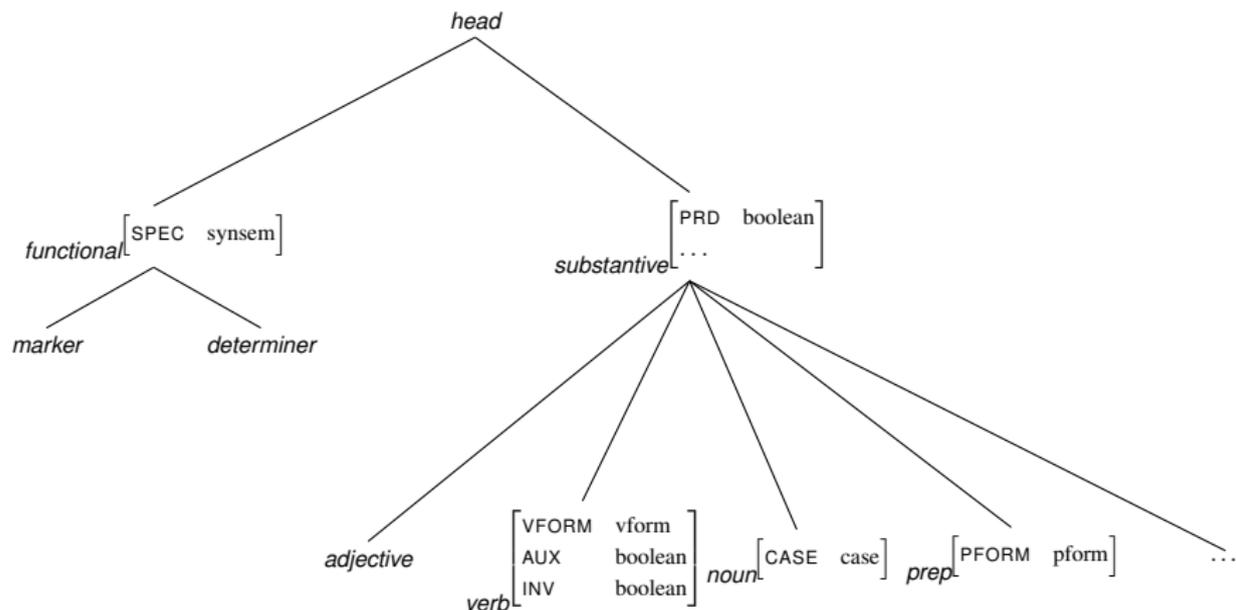
- The sign's syntactic category ("part-of-speech")
 - Given via the feature $\left[\text{HEAD } \textit{head} \right]$, where *head* is the supertype for *noun*, *verb*, *adjective*, *preposition*, *determiner*, *marker*; each of these types selects a particular set of **head features**
- The sign's subcategorization frame/valence, i.e. its potential to combine with other signs to form larger phrases

- Three list-valued features

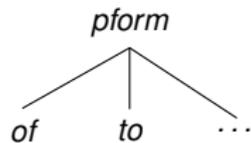
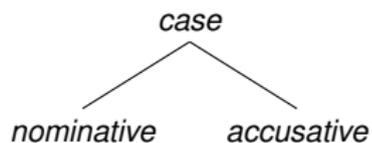
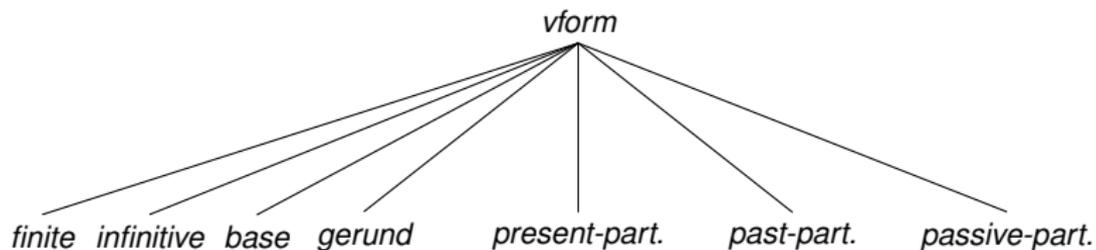
SYNSEM LOC CAT VALENCE	<i>valence</i>	SUBJECT	list(synsem)
		SPECIFIER	list(synsem)
		COMPLEMENTS	list(synsem)

- If any of these lists are non-empty ("*unsaturated*"), the sign has the potential to combine with another sign

Head Information



Features of *head* Types



Valence Features

- The VALENCE lists take as values the list of *synsems* instead of *signs*
- This means that word does not have access to the DTRS list of items on its *valence* lists
- More discussion on different valence lists will follow when we introduce the **valence principle** and **ID schemata**

Semantic Representation

Semantic interpretation of the sign is given as the value to CONTENT

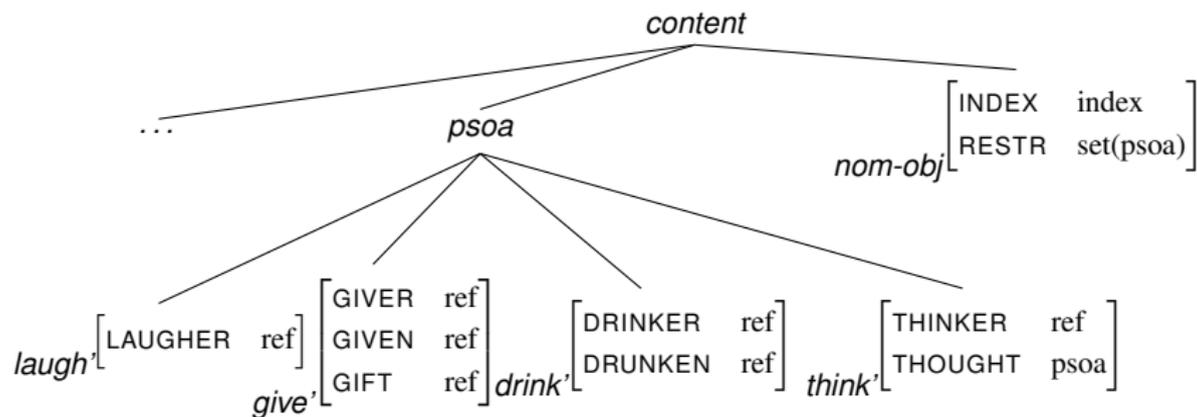
- *nominal-object*: an individual/entity (or a set of them), associated with a referring index, bearing agreement features
- *parameterized-state-of-affairs*: a partial situation; an event relation along with role names for identifying the participants of the event
- *quantifier*: *some, all, every, a, the, ...*
- Note: many of these have been reformulated by “Minimal Recursion Semantics” which allows underspecification of quantifier scopes, though a in-depth discussion of MRS is beyond the scope of this class

Semantic Representation

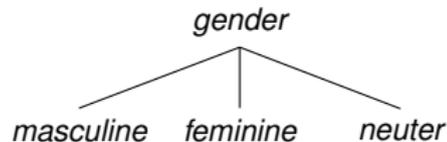
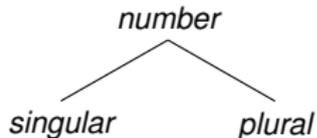
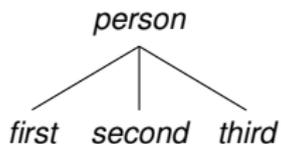
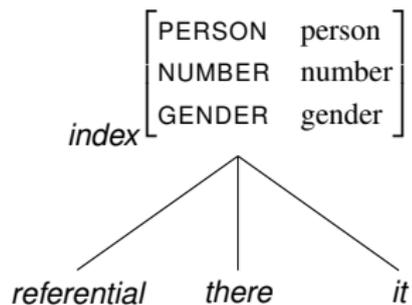
Semantic interpretation of the sign is given as the value to CONTENT

- *nominal-object*: an individual/entity (or a set of them), associated with a referring index, bearing agreement features
- *parameterized-state-of-affairs*: a partial situation; an event relation along with role names for identifying the participants of the event
- *quantifier*: *some, all, every, a, the, ...*
- Note: many of these have been reformulated by “Minimal Recursion Semantics” which allows underspecification of quantifier scopes, though an in-depth discussion of MRS is beyond the scope of this class

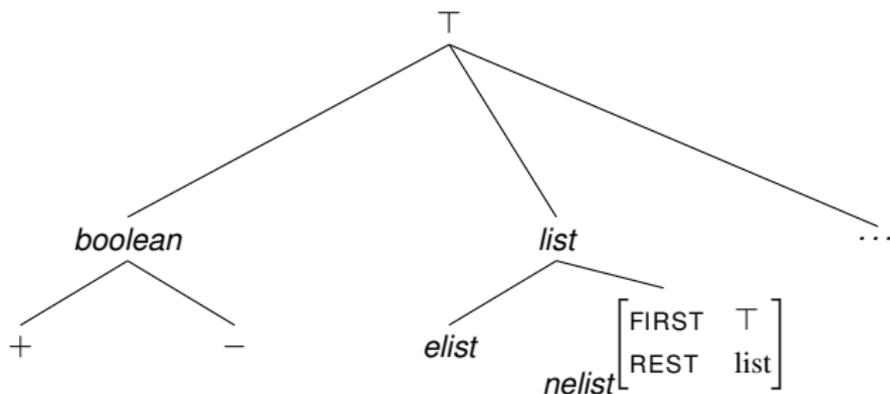
Semantic Representation



Indices



Auxiliary Data Structures

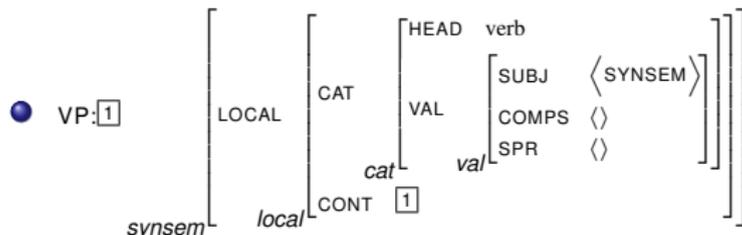
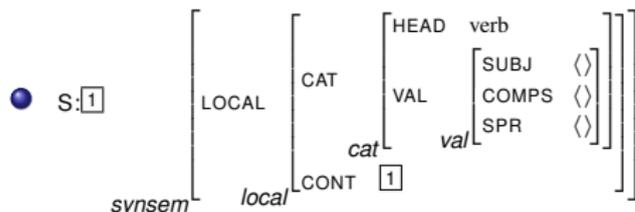
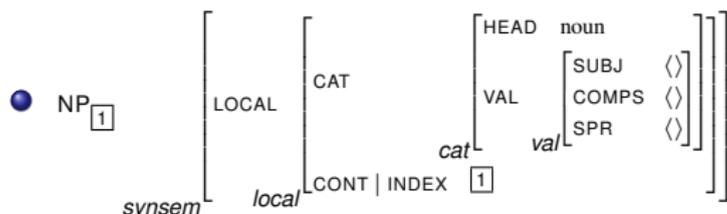


Some List Abbreviations

- Empty list (*elist*) is abbreviated as $\langle \rangle$
- $nelist \left[\begin{array}{l} \text{FIRST} \quad \boxed{1} \\ \text{REST} \quad \boxed{2} \end{array} \right]$ is abbreviated as $\langle \boxed{1} | \boxed{2} \rangle$
- $\langle \dots \boxed{1} | \langle \rangle \rangle$ is equivalent to $\langle \dots \boxed{1} \rangle$
- $nelist \left[\begin{array}{l} \text{FIRST} \quad \boxed{1} \\ \text{REST} \quad \left[\begin{array}{l} \text{FIRST} \quad \boxed{2} \\ \text{REST} \quad \boxed{3} \end{array} \right] \end{array} \right]$ is equivalent to $\langle \boxed{1}, \boxed{2} | \boxed{3} \rangle$
- $\langle \top \rangle$ and $\langle \boxed{1} \rangle$ describe all lists of length one

Abbreviations of Common AVMs

The following abbreviations are used to describe *synsem* objects:



From a linguistic perspective, an HPSG consists of

- A lexicon
licensing basic words
- Lexical rules
licensing derived words
- Immediate dominance (ID) schemata
licensing constituent structure
- Linear precedence (LP) statements
constraining word order
- A set of grammatical principles
expressing generalizations about linguistic objects

The Signature

- Defines the ontology
 - Which kind of objects are distinguished
 - Which properties are modeled
- Consists of
 - Type inheritance hierarchy
 - Appropriate features and constraints on types

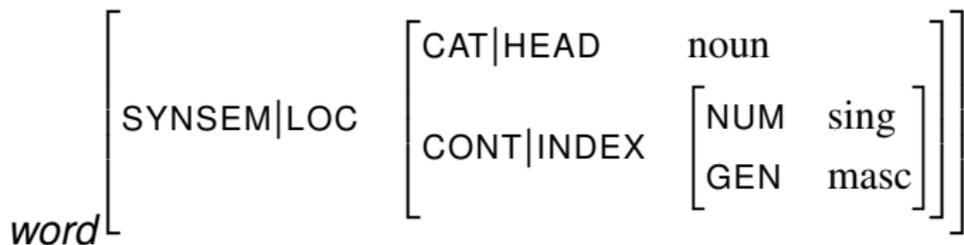
- Linguistic theories are described using AVMs: description language of TFS
- A set of description statements comprises the constraints on what are the admissible linguistic objects (iff there is corresponding well-formed TFS satisfying all the constraints)

Description Example

A verb, for example, can specify that its subject be masculine singular:

(1) Ya spal.
I_{masc.sg} slept_{masc.sg}

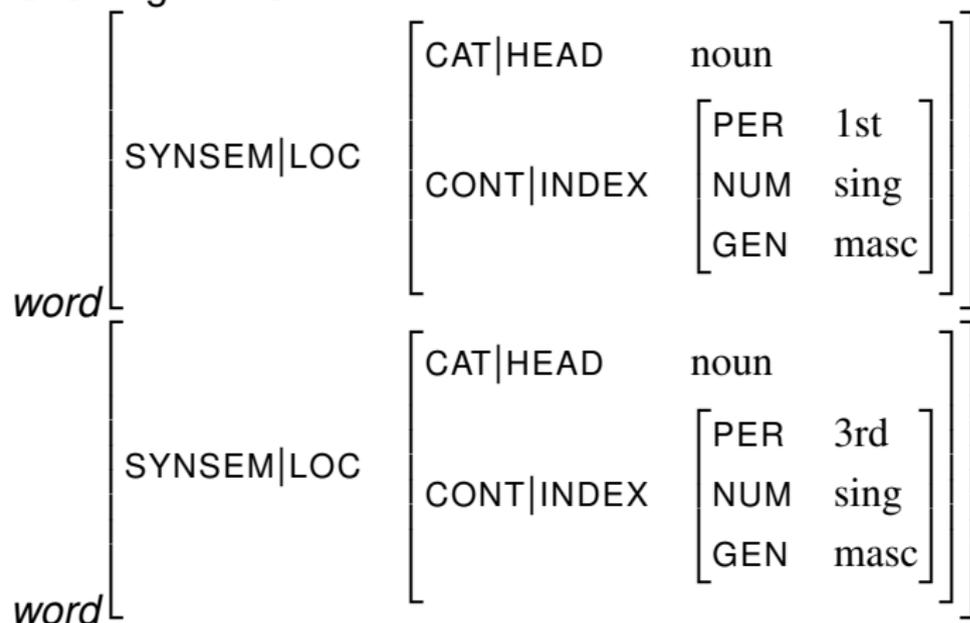
(2) On spal.
He_{masc.sg} slept_{masc.sg}



This AVM specifies the “partial” constraints on the complete (totally well-typed) feature structure of the subject

Subsumption

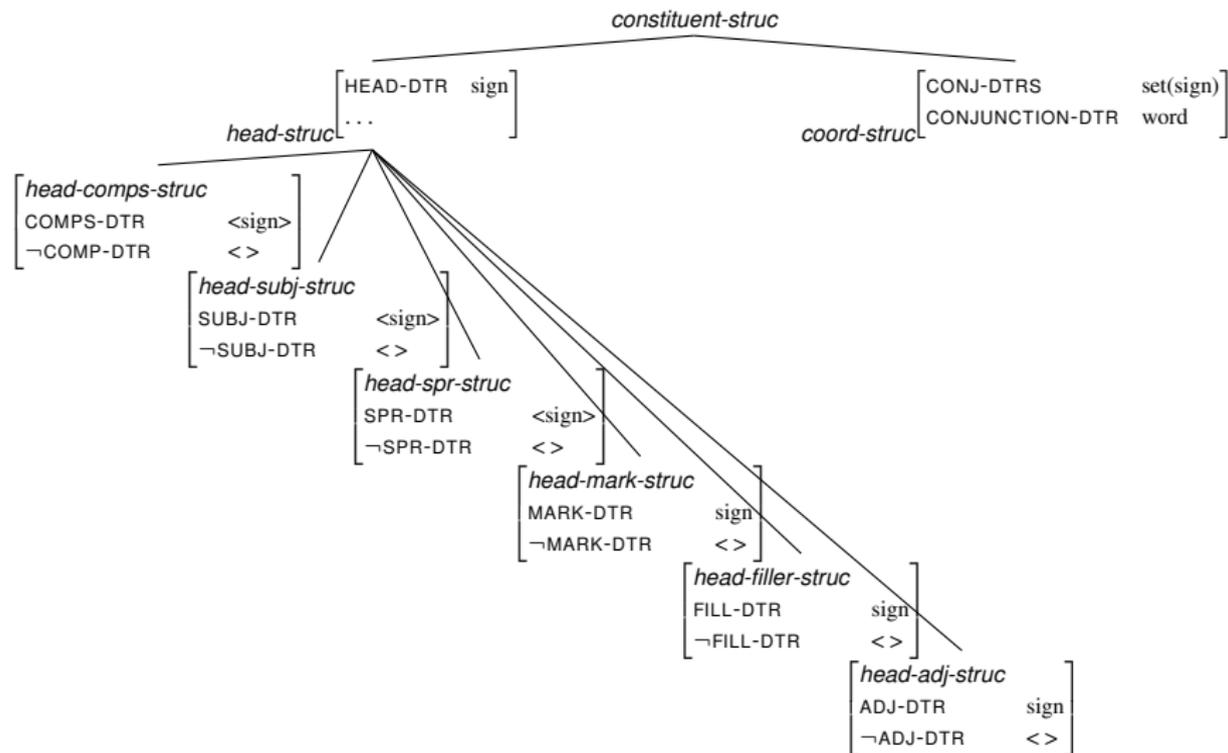
The AVM description on the previous slide subsumes both of the following AVMs



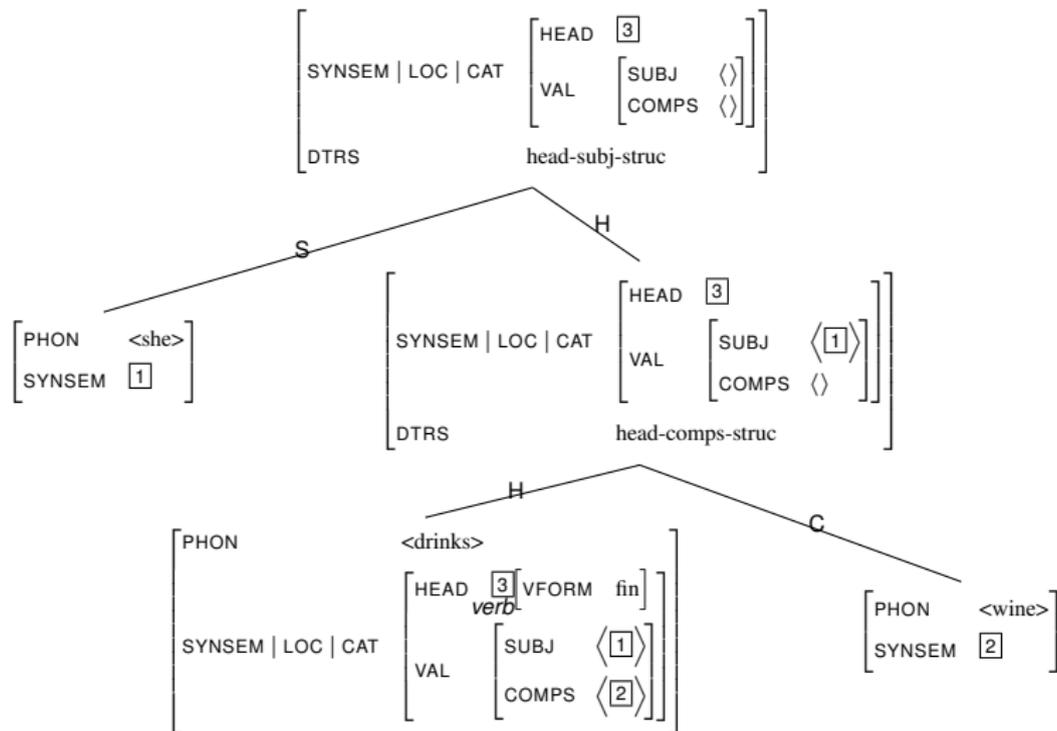
Types of Phrases

- Each *phrase* has a DTRS attribute which has a *constituent-structure* value
- This DTRS value corresponds to what we view in a tree as daughters (with additional grammatical role information, e.g. adjunct, complement, etc.)
- By distinguishing different kinds of *constituent-structures*, we can define different kinds of constructions in a language

An Ontology of Phrases



A Sketch of Head-Subject/Complement Structures



Universal Principles

How exactly did the last example work?

- *drink* has head information specifying that it is a finite verb and subcategories for a subject and an object
 - The head information gets percolated up (the HEAD feature principle)
 - The valence information gets “checked off” as one moves up in the tree (the VALENCE principle)

Such principles are treated as linguistic universals in HPSG

HEAD-Feature Principle

HEAD-feature principle

The value of the HEAD feature of any headed phrase is token-identical with the HEAD value of the head daughter

$$\textit{phrase} \left[\begin{array}{l} \text{DTRS} \quad \text{head-struct} \end{array} \right] \rightarrow \left[\begin{array}{l} \text{SYNSEM} \mid \text{LOC} \mid \text{CAT} \mid \text{HEAD} \quad \boxed{1} \\ \text{DTRS} \mid \text{HEAD-DTR} \mid \text{SYNSEM} \mid \text{LOC} \mid \text{CAT} \mid \text{HEAD} \quad \boxed{1} \end{array} \right]$$

VALENCE Principle

VALENCE principle

In a headed phrase, for each valence feature F , the F value of the head daughter is the concatenation of the phrase's F value with the list of F -DTR's SYNSEM

$$\left[\text{DTRS headed-structure} \right] \rightarrow \left[\begin{array}{l} \text{SYNSEM} \mid \text{LOC} \mid \text{CAT} \mid \text{VAL} \mid F \quad \boxed{1} \\ \text{DTRS} \quad \left[\begin{array}{l} \text{HEAD-DTR} \mid \text{SYNSEM} \mid \text{LOC} \mid \text{CAT} \mid \text{VAL} \mid F \quad \boxed{1} \oplus \langle \boxed{2} \rangle \\ \text{F-DTR} \mid \text{FIRST} \mid \text{SYNSEM} \quad \boxed{2} \end{array} \right] \end{array} \right]$$

- F can be any one of SUBJ, COMPS, SPR
- \oplus stands for list concatenation:

$$\begin{aligned} \text{elist} \oplus \boxed{1} &:= \boxed{1} \\ \langle \boxed{1} \mid \boxed{2} \rangle \oplus \boxed{3} &:= \langle \boxed{1} \mid \boxed{2} \oplus \boxed{3} \rangle \end{aligned}$$

- When the F -DTR is empty, the F valence feature of the head daughter will be copied to the mother phrase

Fallout from These Principles

- Note that agreement is handled neatly, simply by the fact that the SYNSEM values of a word's daughters are token-identical to the items on the VALENCE lists
- How exactly do we decide on a syntactic structure?
- Why the subject is checked off at a higher point in the tree?

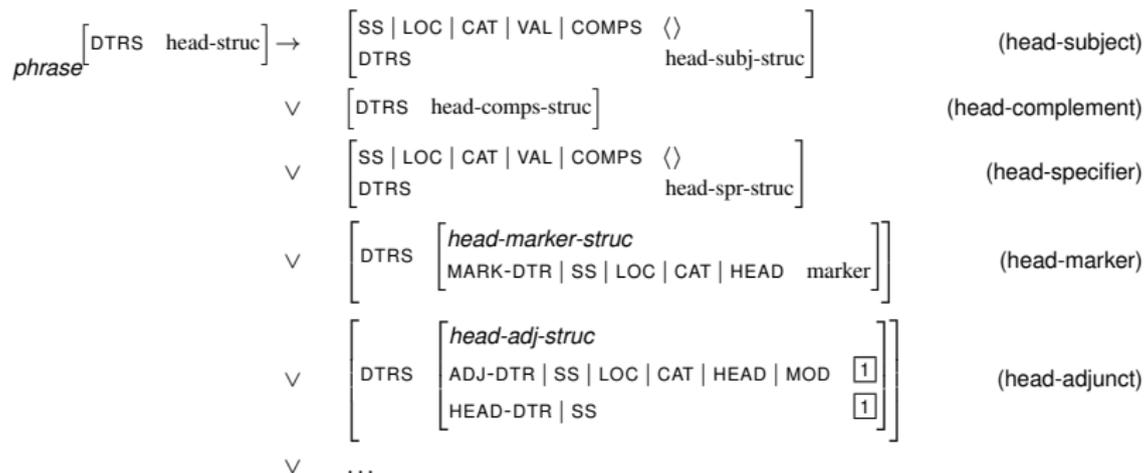
Immediate Dominance (ID) Principle & Schemata

ID Principle

Every headed phrase must satisfy exactly one of the ID schemata

- The exact inventory of valid ID schemata is language specific
- We will introduce a set of ID schemata for English

Immediate Dominance Schemata (for English)



References I



Pollard, C. J. and Sag, I. A. (1994).

Head-Driven Phrase Structure Grammar.

University of Chicago Press, Chicago, USA.