

Syntactic Theory WS09/10

Overview of HPSG Material

February 3, 2010

1 Formalism: Typed Feature Structures

- Token-identity and type-identity (tfs:p.8,9)
- Subsumption relation between typed feature structures (tfs:p.10)
- What is a signature? (tfs:p.14)
- Relation between AVM, TFS and linguistic objects (tfs:p.17)
 - Linguistic objects modeled by feature structures are total in that they are totally well-formed and type-resolved
 - Description of TFS in AVM can be partial
- Unification of typed feature structures (tfs:p.18-19)
- Lists can be defined by a simple type hierarchy (tfs:p.21)

2 Head-Driven Phrase Structure Grammar – Overview

- Try to understand the enlisted properties of HPSG (monostratal, declarative, non-derivational, constraint-based, lexicalist, unification-based) (hpsg1:p.8)
- *Sign* is the basic type for lexical items (words) and phrases. Try to remember the feature geometry (appropriate features and values) of the sign. When you describe a feature, you should have a clear idea about what is the complete path that leads to it. For instance, when you refer to the SPR feature, you know you are referring to the SYNSEM|LOC|CAT|VAL|SPR from the root of the sign. When doing a HPSG analysis, you will draw signs for each terminal (word) and non-terminal (phrase) node. (hpsg1:p.11-14)
- Make sure you understand the list abbreviations like $\langle [] \rangle$, $\langle \boxed{1} \mid \boxed{2} \rangle$ (hpsg1:p.23)
- Make sure you are familiar with the synsem abbreviations like the following:

- NP[acc]_{[1][3rd,sg]}
- VP[inf]⟨_{[1]NP_[2]}⟩:_[3]

(hpsg1:p.24)

- Understand the various HPSG components (hpsg2:p.2)
- Refer to the next session for a list of universal principles
- Make sure you can draw the HPSG analysis in (hpsg2:p.10) by yourself, using the HFP, ValP and the relevant ID schemata
- Understand the constraints in the ID schemata (hpsg2:p.16). For instance, why does the head-subject and head-adjunct schemata require an empty COMPS list on the mother?
- Carefully work on the example shown in (hpsg2:p.6-11), pay special attention to the interaction between principles and schemata
- Lexical rules are introduced in HPSG to capture the systematic variations in lexical information. While the understanding of pluralization and passivization lexical rules are optional and will not be examined, it is important to understand the complement extraction lexical rule (CELR) (udc:p.9)
- For the unbounded dependency constructions in HPSG, only the complement extractions in the strong UDCs will be examined (the rest are optional). You should understand both trace-based and traceless (with lexical rule CELR) analysis. Notice their differences in the ways the slashed local values are introduced. Pay special attention to the interaction between the filler-head schema and the NONLOCAL principle
- Understand the key difference between equi and raising verbs. From semantic point of view, equi verbs systematically assign one more semantic role than their raising counterparts.
- Make sure you understand the treatment of subject equi/raising verbs and object equi/raising verbs in HPSG with different lexical entries. Be able to do HPSG analysis using these lexical entries.

3 List of Principles

For each principle that we introduce, try to practice by reciting their AVM descriptions

- HEAD Feature Principle (HFP) describes the propagation of the HEAD value upward in the tree

- VALENCE Principle (ValP) describes the relation between the valence lists of the parent and daughters
- ID Principle says that the immediate dominance relations are described by ID schemata. Refer to the next section for further details
- Semantic Principle (SemP) describes the identification of the semantic head
- SPEC Principle (SPEC) describes the identification of the SPEC feature for functional signs
- MARKING Principle describes the identification of the MARKING feature
- NONLOCAL Principle describes how the NONLOCAL feature manages the slashed elements

4 List of ID Schemata for English

- head-subject-schema
- head-complement-schema
- head-specifier-schema
- head-adjunct-schema
- head-marker-schema
- head-filler-schema
- head-subject-complement-schema

5 An Incomplete List of Features

- ADJUNCT-DAUGHTER (ADJ-DTR)
- ARGUMENT (ARG)
- AUXILIARY (AUX)
- BACKGROUND (BACKGR, BG)
- CASE
- CATEGORY (CAT)
- COMPLEMENTS (COMPS)
- COMPLEMENTS-DAUGHTER (COMPS-DTR)

- CONTENT (CONT)
- CONTEXT (CTXT)
- DAUGHTERS (DTRS)
- FILLER-DAUGHTER (FILL-DTR)
- FIRST
- GENDER (GEND, GEN, G)
- HEAD (H)
- HEAD-DAUGHTER (HD-DTR, HD)
- INDEX (IDX)
- INVERTED (INV)
- INHERITED (INHER)
- INSTANCE (INST)
- LOCAL (LOC)
- MARKER-DAUGHTER (MARK-DTR)
- MARKING
- MODIFIED (MOD)
- NONHEAD-DAUGHTER (NONHEAD-DTR, NH-DTR)
- NONLOCAL (NONLOC)
- NUMBER (NUM, N)
- PERSON (PER, P)
- PFORM
- PHONOLOGY (PHON)
- PREDICATIVE (PRD)
- REST
- RESTRICTION (RESTR)
- SLASH
- SOA-ARG
- SPECIFIED (SPEC)

- SPECIFIER (SPR)
- SPECIFIER-DAUGHTER (SPR-DTR)
- SUBJECT (SUBJ)
- SUBJECT-DAUGHTER (SUBJ-DTR)
- SYNSEM (SS)
- TO-BIND (TO-B)
- VALENCE (VAL)
- VFORM

6 An Incomplete List of Types

- *accusative (acc)*
- *adjective (adj)*
- *adverb (adv)*
- *base (bse)*
- *boolean (bool)*
- *category (cat)*
- *determiner (det)*
- *elist*
- *false (-)*
- *feminine (fem, f)*
- *finite (fin)*
- *functional (func)*
- *gender (gend, gen, g)*
- *gerund*
- *head*
- *index*
- *infinite (inf)*
- *it*

- *list, list(σ)*
- *local (loc)*
- *marker*
- *masculine (masc, m)*
- *nelist*
- *neuter (neut, n)*
- *nominal-object (nom-obj)*
- *nominative (nom)*
- *nonlocal (nonloc)*
- *noun*
- *number (num)*
- *passive-participle (psp)*
- *past-participle (pst)*
- *person (per)*
- *phrase (ph)*
- *plural (pl)*
- *preposition (prep)*
- *psoa*
- *ref*
- *set, set(σ)*
- *sign*
- *singular (sg)*
- *substantive (subst)*
- *synsem (ss)*
- *there*
- *top (\top)*
- *true (+)*
- *unmarked*

- *verb*
- *word*
- *1st*
- *2nd*
- *3rd*