

# LG611: HPSG Basic Ideas and Principles

Doug Arnold  
doug@essex.ac.uk

## 1 Historical Background

- FUG: Feature (AV) structures.
- GPSG, Head Grammars: Unbounded Dependencies, LP (but: no ‘rules’; different view of subcategorization)
- LFG: Lexicalism (e.g. lexical treatment of passive) (but: sorted/typed FSs; Sign Based)
- CG, UCG: treatment of adjuncts, subcategorization.
- GB: Principles, Binding Theory.

## 2 Basic Ideas

The Standard Ideas of Generative Grammar (Competence, Performance, Universals, Language Acquisition).

- Formalization.
- Lexicalism.
- Principle (vs. Rule) Based.
- Sign Based (integrated theory of syntax, semantics, and phonology).
- Uniform Representation for principles, ‘rules’, syntactic structures, and lexical entries.
- Use of Typed Feature Structures.
- “Unification Based”.
- Constraint Based (Non-Derivational).
- Phrase Structure.

## 3 Formal Foundations

### 3.1 Feature Structures

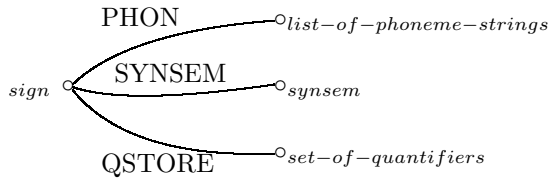
Linguistic information is modelled by means of *feature structures*.

- partial information
- recursivity
- re-entrance
- subsumption (extension; unification, generalization)

### 3.2 Notation, etc.

Feature Structures (FSs) model linguistic entities.

- Formally, FSs are interpreted as Deterministic Finite State Automata (pictured as graphs)
- FSs are described by sets of equations.
- FSs are represented as AV matrices.



$rooted - at(x0) \wedge sign(x0) \wedge$   
 $PHON(x0, x1) \wedge list - of - phoneme - strings(x1) \wedge$   
 $SYNSEM(x0, x2) \wedge synsem(x2) \wedge QSTORE(x0, x3)$   
 $\wedge set - of - quantifiers(x3) \wedge \dots$

$sign \left[ \begin{array}{ll} PHON & \boxed{1} \text{ list-of-phoneme-strings} \\ SYNSEM & \boxed{2} \text{ synsem} \\ QSTORE & \boxed{3} \text{ set-of-quantifiers} \end{array} \right]$

### 3.3 Unification

$$A \left[ \begin{array}{l} 1 \\ V_1 \end{array} \right] \sqcup_A \left[ \begin{array}{l} 2 \\ V_2 \end{array} \right] = \left[ \begin{array}{l} 1 \\ V_1 \\ A_2 \\ V_2 \end{array} \right]$$

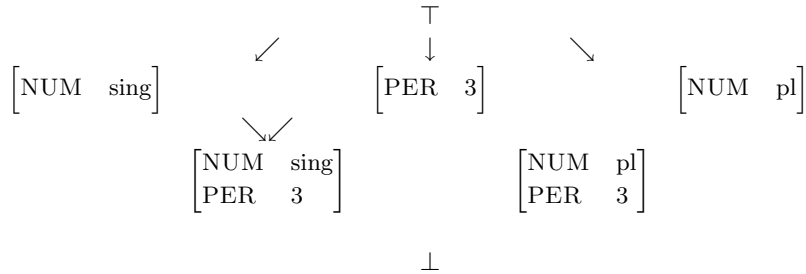
$$A \left[ \begin{array}{l} 1 \\ V_1 \end{array} \right] \sqcup_A \left[ \begin{array}{l} 1 \\ V_2 \end{array} \right] = \perp$$

### 3.4 Terminology, General Properties of Unification

$A \sqsubseteq B$      $A$  subsumes  $B$   
 $B \supseteq A$      $B$  extends  $A$

idempotency     $A \sqcup A = A$   
 commutativity     $A \sqcup B = B \sqcup A$   
 associativity     $(A \sqcup B) \sqcup C = A \sqcup (B \sqcup C)$   
 top     $\top \sqcup A = A$   
 botton     $\perp \sqcup A = \perp$   
 interdefinability     $A \sqsubseteq B \text{ iff } A \sqcup B = B$

### 3.5 Subsumption Lattice



### 3.6 Extensions

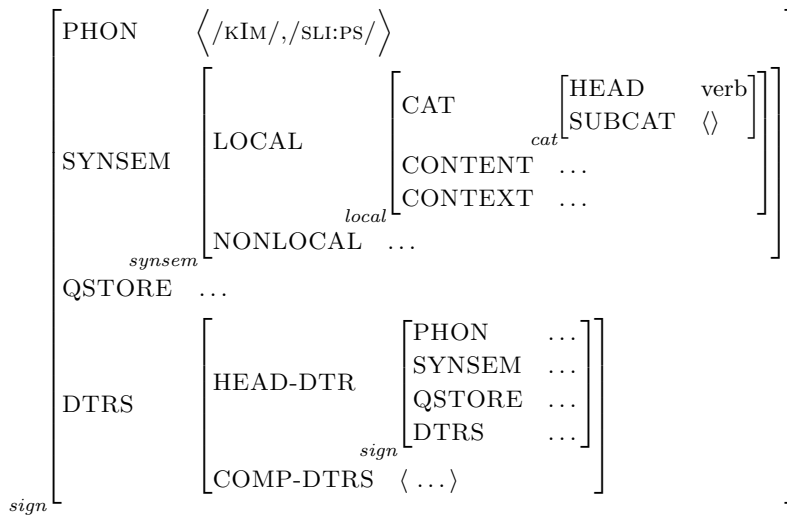
- Types
- Boolean Constraints
  1. disjunction
  2. implication
  3. negation
- Lists (e.g. SUBCAT)
- Sets (e.g. DTRS)
- Functionally dependent values (e.g. append, union)

Grammars are combinations of conditional FSs:

1.  $UG = P_1 \wedge P_2 \wedge \dots P_n$
2.  $Grammar_{English} = R_1 \vee R_2 \dots R_n$
3.  $Lexicon_{English} = L_1 \vee L_2 \vee \dots L_n$
4.  $English = P_1 \wedge P_2 \wedge \dots P_n \wedge ( R_1 \vee R_2 \vee \dots R_n \vee L_1 \vee L_2 \vee \dots L_n )$

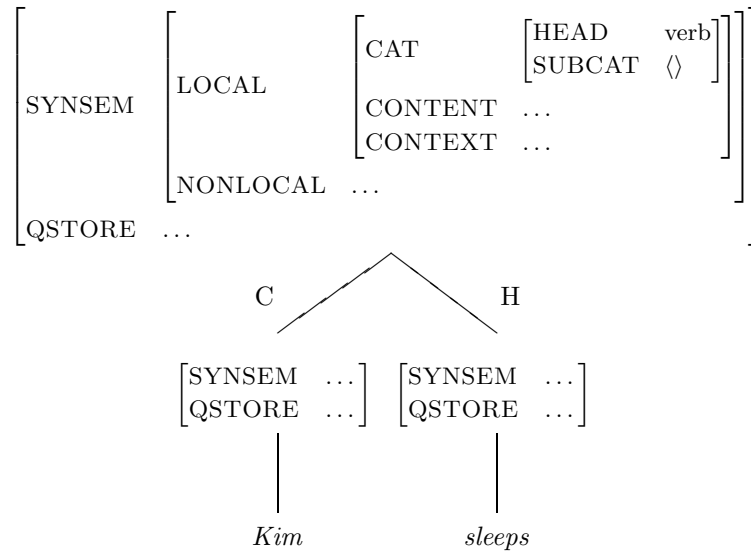
## 4 Top Level Structure of Signs

(1) Kim sleeps.



Standard tree notation is often used:

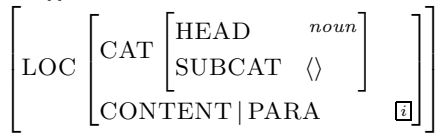
- DTRs can be written as daughters under the main sign (branches should be labelled to indicate what sort of DTR is involved);
- PHON values can be written as leaves (the PHON of a sign is assumed to be the concatenation of the PHON values of the daughters, in the order given).



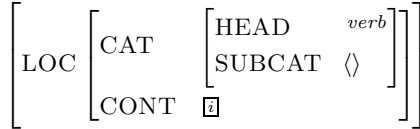
If we ignore QSTORE, then nodes are just labelled with SYNSEM values.

## 5 Abbreviations

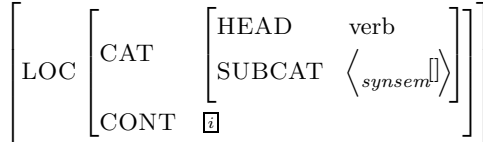
- NP<sub>[i]</sub> abbreviates



- S:<sub>[i]</sub> abbreviates



- VP:<sub>[i]</sub> abbreviates



## 6 Principles

**Subcategorization Principle** : In a headed phrase (i.e. a phrasal sign whose DTRS value is of sort *head-struct*), the SUBCAT value of the head daughter is the concatenation of the phrases SUBCAT list with the list (in order of increasing obliqueness) of the SYNSEM values of the complement daughters.

**Head Feature Principle (HFP)**: The HEAD value of any headed phrase is structure-shared with the HEAD value of the head daughter.

**SPEC principle** : If a non-head daughter in a headed structure bears a SPEC value, it is token identical to the SYNSEM value of the head daughter.

**Semantics Principle** : In a headed phrase, the CONTENT value is token identical to that of the adjunct daughter if the DTRS value is of sort *head-adj-struct*, and with that of the head

daughter otherwise.  
**Quantifier Inheritance Principle** : ...