Grammar: Features and Unification

Plan for the Talk

- Problems with CFG (PCFG)
- Features Structure
  - Attribute-value Matrix (AVM)
- Unification
- Grammar formalisms based on unification

Agreement

- Constraints that hold among various constituents.
- For example, in English, determiners and the head nouns in NPs have to agree in their number.

- Which of the following cannot be parsed by the rule

NP → Det Nominal **?**

(O) This flight
(O) Those flights

(X) This flights
(X) Those flight

Agreement

- Constraints that hold among various constituents.
- For example, in English, determiners and the head nouns in NPs have to agree in their number.

- Which of the following cannot be parsed by the rule

NP $\longrightarrow$ Det Nominal **?**

➔ This rule does not handle agreement! (The rule does not detect whether the agreement is correct or not.)

(O) This flight	(X) This flights
(O) Those flights	(X) Those flight

Problem with CFG/PCFG

- Our earlier NP rules are clearly deficient since they don't capture the agreement constraint
  - *NP $\longrightarrow$ Det Nominal*
    - Accepts, and assigns correct structures, to grammatical examples (*this flight*)
    - But its also happy with incorrect examples (*these flight)
  - Such a rule is said to ***overgenerate***.
  - We'll come back to this in a bit

Verb Phrases

- English *VPs* consist of a head verb along with 0 or more following constituents which we'll call ***arguments***.

VP	$\rightarrow$	$Verb$	disappear
VP	$\rightarrow$	$Verb\ NP$	prefer a morning flight
VP	$\rightarrow$	$Verb\ NP\ PP$	leave Boston in the morning
VP	$\rightarrow$	$Verb\ PP$	leaving on Thursday

Subcategorization

- ***John sneezed the book**
- ***I prefer United has a flight**
- ***Give with a flight**

- As with agreement phenomena, we need a way to formally express the constraints!

Subcategorization

- Sneeze: John sneezed
- Find: Please find [a flight to NY]$_{NP}$
- Give: Give [me]$_{NP}$[a cheaper fare]$_{NP}$
- Help: Can you help [me]$_{NP}$[with a flight]$_{PP}$
- Prefer: I prefer [to leave earlier]$_{TO\text{-}VP}$
- Told: I was told [United has a flight]$_{S}$
- …

Subcategorization

- But, even though there are many valid VP rules in English, not all verbs are allowed to participate in all those VP rules.
- We can subcategorize the verbs in a language according to the sets of VP rules that they participate in.
- This is a modern take on the traditional notion of transitive/intransitive.
- Modern grammars may have 100s or such classes.

Problem with CFG/PCFG

- Right now, the various rules for VPs ***overgenerate***.
  - They permit the presence of strings containing verbs and arguments that don't go together
  - For example
  - VP -> V NP therefore

    Sneezed the book is a VP since "sneeze" is a verb and "the book" is a valid NP

Possible CFG Solution

- Possible solution for agreement.
- Can use the same trick for all the verb/VP classes.

- SgS -> SgNP SgVP
- PlS -> PlNp PlVP

- SgNP -> SgDet SgNom
- PlNP -> PlDet PlNom

- PlVP -> PlV NP
- SgVP ->SgV Np
- …

CFG Solution for Agreement

- Pro:
  - It works and stays within the power of CFGs
- Con:
  - loss of generalization – “apple” and “apples” are treated as if they are two separate words
  - And it doesn’t scale all that well because of the interaction among the various constraints explodes the number of rules in our grammar.

Non-CFG Solution for Agreement

- Add “constraints” to each rule
  - S -> NP VP

    constraint: only if the number of NP is equal to the number of the VP

- Instead of replicating rules…
  - SgS -> SgNP SgVP
  - PlS -> PlNp PlVP
  - SgNP -> SgDet SgNom
  - PlNP -> PlDet PlNom
  - PlVP -> PlV NP
  - SgVP ->SgV Np
  - …

Plan for the Talk

- Problems with CFG (PCFG)
- Features Structure
  - Attribute-value Matrix (AVM)
- Unification
- Grammar formalisms based on unification

Feature Structure

- "Features" in formal grammar

$\neq$

- "Features" in machine learning

- Attribute-value Matrix (AVM)
  - Feature Path
  - Reentrant structure

Feature Structure

This feature structure is used in many grammar formalism that goes beyond CFG, such as

- Head-Driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1987, 1994)
- Lexical Functional Grammar (LFG) (Bresnan, 1982)
- Construction Grammar (Kay and Fillmore, 1999)
- Unification Categorial Grammar (Uszkoreit, 1986)

Attribute-value matrix (AVM)

Definition:

$$\begin{pmatrix} \text{FEATURE\_1} & \text{value\_1} \\ \text{FEATURE\_2} & \text{value\_2} \\ \text{....} & \\ \text{FEATURE\_n} & \text{value\_n} \end{pmatrix}$$

For example:

$$\begin{pmatrix} \text{NUMBER} & \text{sg} \end{pmatrix}$$

Attribute-value matrix (AVM)

More Examples:

$$\begin{bmatrix} \text{CAT} & \text{NP} \\ \text{NUMBER} & \text{sg} \\ \text{PERSON} & \text{3rd} \end{bmatrix}$$

Attribute-value matrix (AVM)

Hierarchical Structure: “value” can be another AVM object

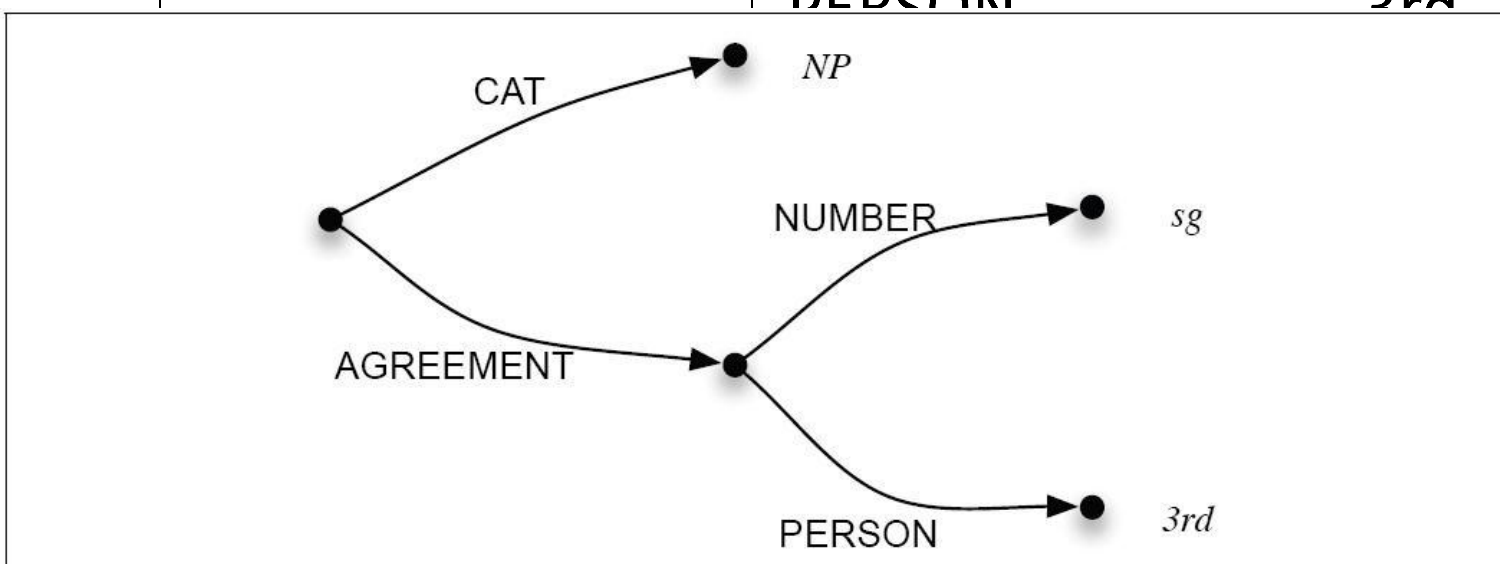
$$
\begin{bmatrix}
\text{CAT} & \text{NP} \\
\text{NUMBER} & \text{sg} \\
\text{PERSON} & \text{3rd}
\end{bmatrix}
$$

$$
\begin{bmatrix}
\text{CAT} & \text{NP} \\
\text{AGREEMENT} & \begin{bmatrix} \text{NUMBER} & \text{sg} \\ \text{PERSON} & \text{3rd} \end{bmatrix}
\end{bmatrix}
$$

Feature Path

Feature Path: a sequence of features in the feature structure (AVM) leading to a particular value

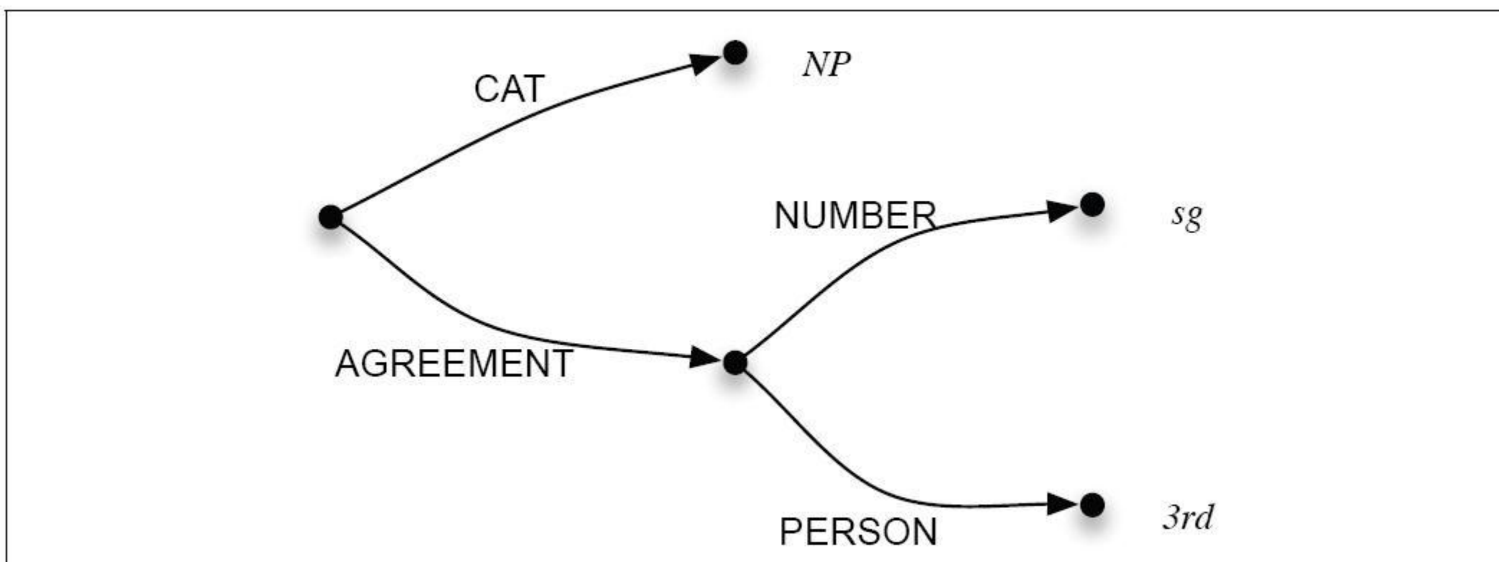
$$\begin{bmatrix} \text{CAT} & \text{NP} \\ \text{AGREEMENT} & \begin{bmatrix} \text{NUMBER} & \text{sg} \\ \text{PERSON} & \text{3rd} \end{bmatrix} \end{bmatrix}$$



Feature Path

Feature Path: a sequence of features in the feature structure (AVM) leading to a particular value

$$\begin{bmatrix} \text{CAT} & \text{NP} \\ \text{AGREEMENT} & \begin{bmatrix} \text{NUMBER} & \text{sg} \\ \text{PERSON} & \text{3rd} \end{bmatrix} \end{bmatrix}$$



Attribute-value matrix (AVM)

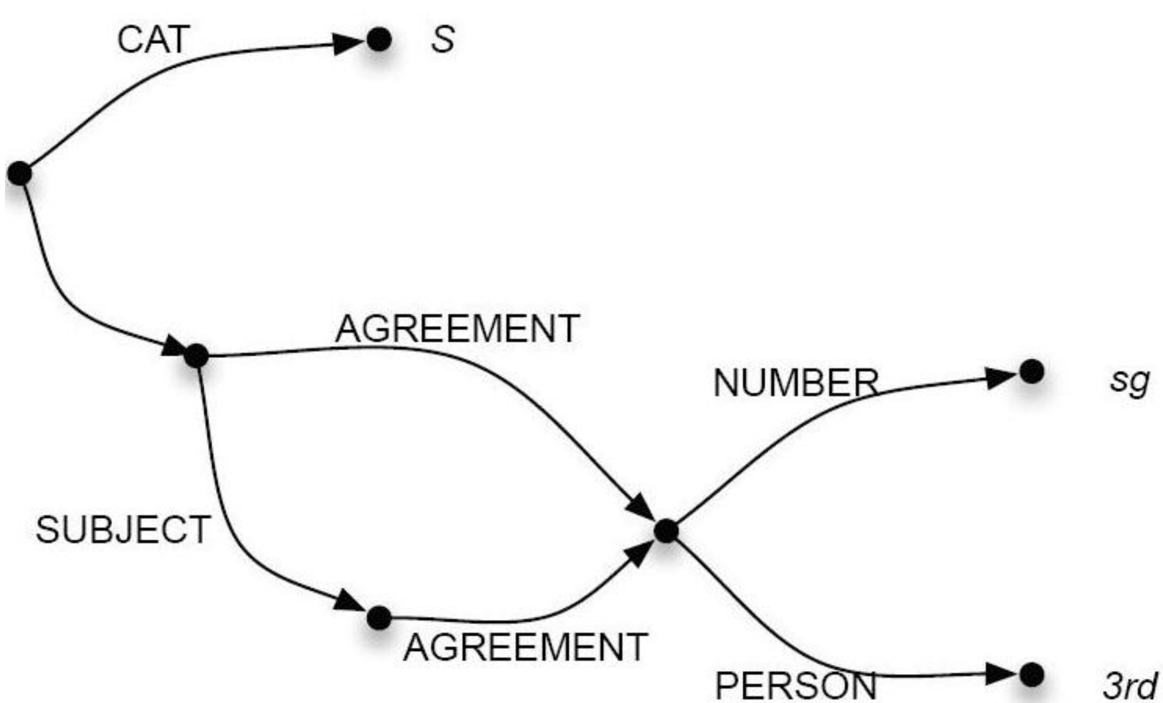
Reentrant Structure:

$$\begin{bmatrix} \text{CAT} & \text{S} \\ \text{HEAD} & \begin{bmatrix} \text{AGREEMENT } \textbf{[1]} & \begin{bmatrix} \text{NUMBER} & \text{sg} \\ \text{PERSON} & 3^{rd} \end{bmatrix} \\ \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & \textbf{[1]} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Reentrant Structure:

$$\begin{bmatrix} \text{CAT} & \text{S} \\ \text{HEAD} & \begin{bmatrix} \text{AGREEMENT} \ \text{[1]} & \begin{bmatrix} \text{NUMBER} & \text{sg} \\ \text{PERSON} & 3^{rd} \end{bmatrix} \\ \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & \text{[1]} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Feature Path:



Feature Structure

- "Features" in formal grammar

$\neq$

- "Features" in machine learning

- Attribute-value Matrix (AVM)
  - Feature Path
  - Reentrant structure

- This feature structure is used in many grammar formalism that goes beyond CFG, such as HPSG, LFG

Plan for the Talk

- Problems with CFG (PCFG)
- Features Structure
  - Attribute-value Matrix (AVM)
- Unification
- Grammar formalisms based on unification

Unification of Feature Structure

- Unification of two feature structure (AVM) finds the most general feature structure that is compatible with the two given AVMs.
- [NUMBER sg] U [NUMBER sg] =
- [NUMBER sg] U [NUMBER pl] =
- [NUMBER sg] U [NUMBER [ ]] =

Unification of Feature Structure

- Unification of two feature structure (AVM) finds the most general feature structure that is compatible with the two given AVMs.

- [NUMBER sg] U [NUMBER sg] = [NUMBER sg]

- [NUMBER sg] U [NUMBER pl] ➔ Fails !

- [NUMBER sg] U [NUMBER [ ]] = [NUMBER sg]

Unification of Feature Structure

- Unification of two feature structure (AVM) finds the most general feature structure that is compatible with the two given AVMs.

- [NUMBER sg] U [PERSON 3rd] =

Unification of Feature Structure

- Unification of two feature structure (AVM) finds the most general feature structure that is compatible with the two given AVMs.

- [NUMBER sg] U [PERSON 3rd] = $\begin{pmatrix} \text{NUMBER sg} \\ \text{PERSON } 3^{rd} \\ \text{CATEGORY NP} \end{pmatrix}$?

Unification of Feature Structure

- Unification of two feature structure (AVM) finds the <u>most general</u> feature structure that is compatible with the two given AVMs.

- [NUMBER sg] U [PERSON 3rd] = $\begin{pmatrix} \text{NUMBER sg} \\ \text{PERSON 3}^{rd} \\ \cancel{\text{CATEGORY NP}} \end{pmatrix}$?

Unification of Feature Structure

- Unification of two feature structure (AVM) finds the most general feature structure that is compatible with the two given AVMs.

- [NUMBER sg] U [PERSON 3rd] = $\begin{bmatrix} \text{NUMBER sg} \\ \text{PERSON 3rd} \end{bmatrix}$

Unification of Feature Structure

$$
\begin{bmatrix}
\text{AGREEMENT } [1] & \begin{bmatrix} \text{NUMBER} & \text{sg} \\ \text{PERSON} & 3^{rd} \end{bmatrix} \\
\text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & [1] \end{bmatrix}
\end{bmatrix}
$$

$$
\text{U} \quad \begin{bmatrix} \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{PERSON} & 3^{rd} \\ \text{NUMBER} & \text{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix}
$$

=

Unification of Feature Structure

$$
\begin{bmatrix}
\text{AGREEMENT } [1] & \begin{bmatrix} \text{NUMBER} & \text{sg} \\ \text{PERSON} & 3^{rd} \end{bmatrix} \\
\text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT } [1] \end{bmatrix}
\end{bmatrix}
$$

$$
\text{U} \quad
\begin{bmatrix}
\text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{PERSON} & 3^{rd} \\ \text{NUMBER} & \text{sg} \end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

$$
= \quad
\begin{bmatrix}
\text{AGREEMENT } [1] & \begin{bmatrix} \text{NUMBER} & \text{sg} \\ \text{PERSON} & 3^{rd} \end{bmatrix} \\
\text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT } [1] \end{bmatrix}
\end{bmatrix}
$$

Unification of Feature Structure

$$\begin{bmatrix} \text{AGREEMENT} & [1] \\ \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & [1] \end{bmatrix} \end{bmatrix}$$

$$\text{U} \quad \begin{bmatrix} \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{PERSON} & 3^{rd} \\ \text{NUMBER} & \text{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$=$$

Unification of Feature Structure

$$\begin{bmatrix} \text{AGREEMENT} \ [1] & \\ \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} \ [1] \end{bmatrix} \end{bmatrix}$$

$$\text{U} \quad \begin{bmatrix} \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{PERSON} & 3^{rd} \\ \text{NUMBER} & \text{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} \text{AGREEMENT} & [1] \\ \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} \ [1] & \begin{bmatrix} \text{PERSON} & 3^{rd} \\ \text{NUMBER} & \text{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Unification of Feature Structure

$$\begin{bmatrix} \text{AGREEMENT } [1] & \begin{bmatrix} \text{NUMBER} & \text{sg} \\ \text{PERSON} & 3^{rd} \end{bmatrix} \\ \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & [1] \end{bmatrix} \end{bmatrix}$$

$$\cup \quad \begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{NUMBER} & \text{sg} \\ \text{PERSON} & 3^{rd} \end{bmatrix} \\ \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{PERSON} & 3^{rd} \\ \text{NUMBER} & \text{pl} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$=$$

Unification of Feature Structure

$$\begin{bmatrix} \text{AGREEMENT} & [1] \begin{bmatrix} \text{NUMBER} & \text{sg} \\ \text{PERSON} & 3^{rd} \end{bmatrix} \\ \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & [1] \end{bmatrix} \end{bmatrix}$$

$$\text{U} \quad \begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{NUMBER} & \text{sg} \\ \text{PERSON} & 3^{rd} \end{bmatrix} \\ \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{PERSON} & 3^{rd} \\ \text{NUMBER} & \text{pl} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Fails!

Unification of Feature Structure

$$\begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{NUMBER} & sg \end{bmatrix} \\ \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{NUMBER} & sg \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\text{U} \quad \begin{bmatrix} \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{PERSON} & 3^{rd} \\ \text{NUMBER} & sg \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Unification of Feature Structure

$$
\begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{NUMBER} & \text{sg} \end{bmatrix} \\ \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{NUMBER} & \text{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix}
$$

$$
\text{U} \quad \begin{bmatrix} \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{PERSON} & 3^{rd} \\ \text{NUMBER} & \text{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix}
$$

$$
= \begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{NUMBER} & \text{sg} \end{bmatrix} \\ \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{PERSON} & 3^{rd} \\ \text{NUMBER} & \text{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix}
$$

Plan for the Talk

- Problems with CFG (PCFG)
- Features Structure
  - Attribute-value Matrix (AVM)
- Unification
- Grammar formalisms based on unification

Grammar Theories based on Unification

- Head-Driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1987, 1994)
- Lexical Functional Grammar (LFG) (Bresnan, 1982)
- Construction Grammar (Kay and Fillmore, 1999)
- Unification Categorial Grammar (Uszkoreit, 1986)

- Note that these grammar formalisms tend to focus on illuminating syntactic analysis, rather than providing computational implementations. (computationally very expensive)

Example of AVM used in HPSG

- Head-Driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1987, 1994)
  - Non-derivational
  - Constraint-based
  - Highly lexicalized

- Each word is fully described with
  - morpho-syntactic features
  - semantic features

Example of AVM used in HPSG

- “put” --- e.g., “John put a book on the table”

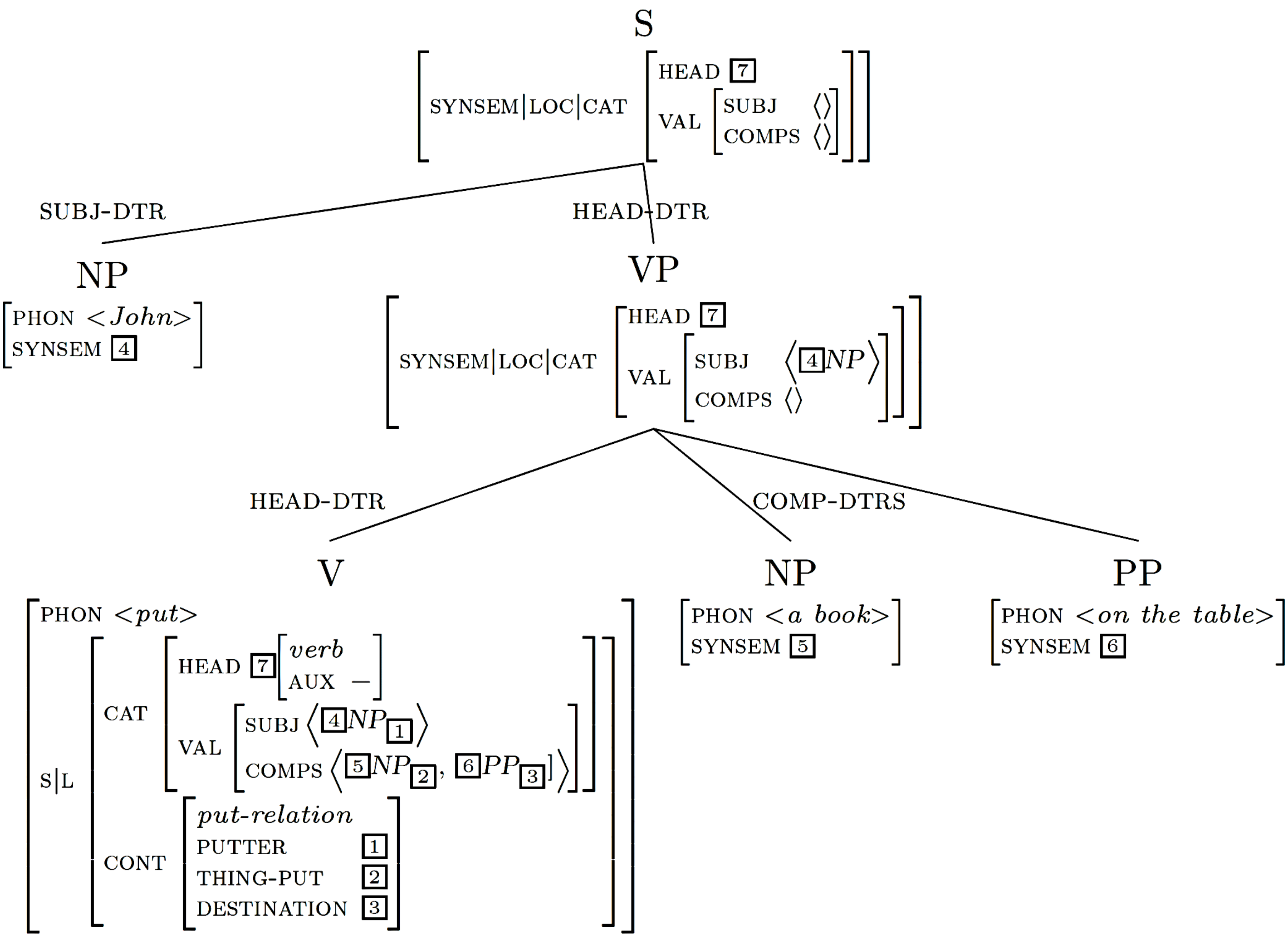
$$\left[\begin{array}{ll} \textit{word} & \\ \text{PHON} & <\textit{put}> \\ \text{SYNSEM}|\text{LOCAL} & \left[\begin{array}{ll} \text{CATEGORY} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \textit{verb} & \\ \text{AUX} & - \end{array}\right] \\ \text{VALENCE} & \left[\begin{array}{ll} \text{SUBJ} & \langle NP_{\boxed{1}} \rangle \\ \text{COMPS} & \langle NP_{\boxed{2}}, PP_{\boxed{3}} \rangle \end{array}\right] \end{array}\right] \\ \text{CONTENT} & \left[\begin{array}{ll} \textit{put-relation} & \\ \text{PUTTER} & \boxed{1} \\ \text{THING-PUT} & \boxed{2} \\ \text{DESTINATION} & \boxed{3} \end{array}\right] \end{array}\right] \end{array}\right]$$

Example of AVM used in HPSG

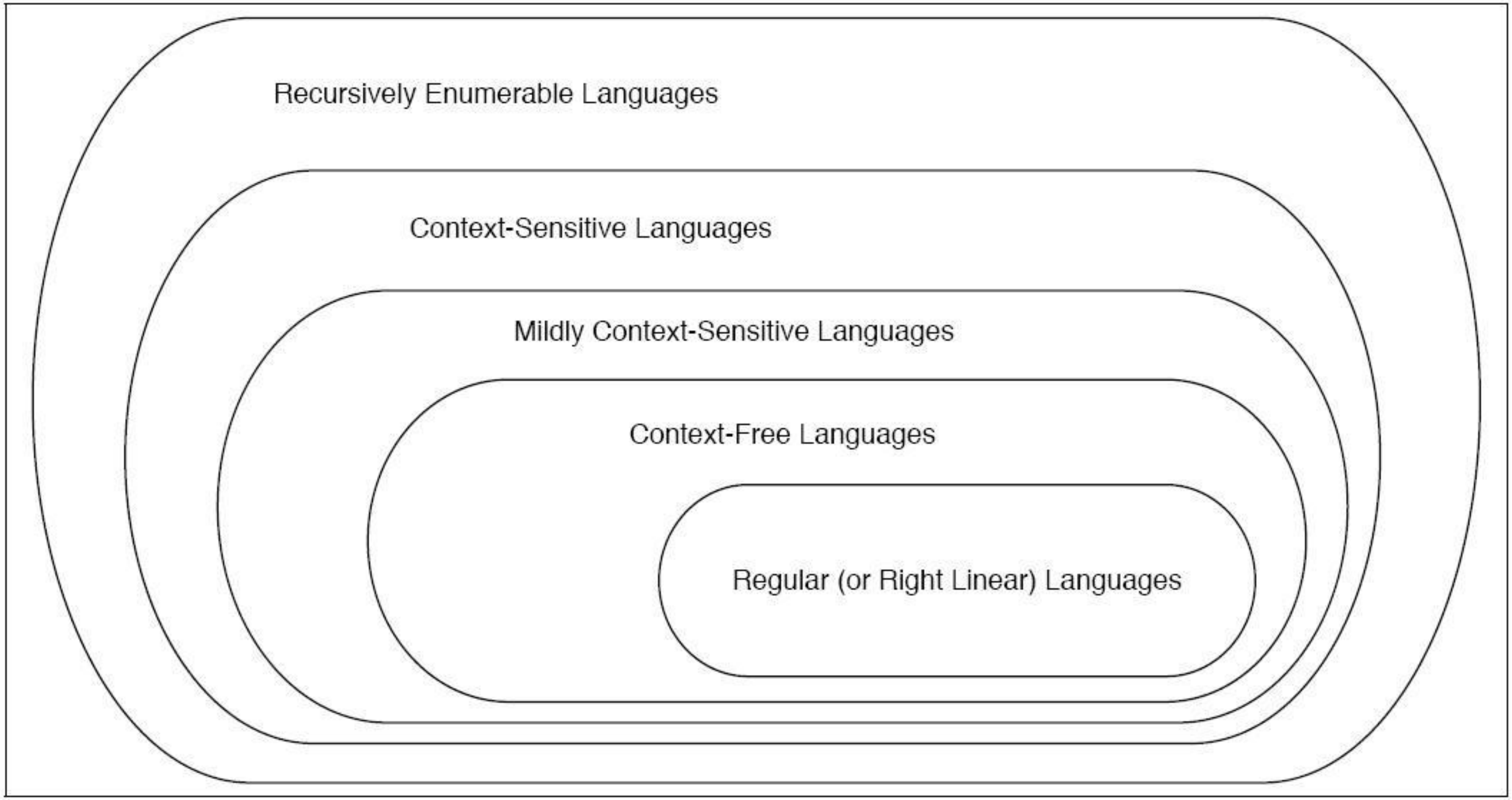
- Each word can have many different AVM descriptions (due to polysemy, or multiple possible syntactic relations with other words/phrases)
- each lexical_entry corresponds to an AVM description such as shown below:

word → lexical_entry_1
V lexical_entry_2
V …
V lexical_entry_n

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & <\textit{put}> \\ \text{SYNSEM}|\text{LOCAL} & \begin{bmatrix} \text{CATEGORY} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AUX} & - \end{bmatrix} \\ \text{VALENCE} & \begin{bmatrix} \text{SUBJ} & \langle NP_{\boxed{1}} \rangle \\ \text{COMPS} & \langle NP_{\boxed{2}}, PP_{\boxed{3}} \rangle \end{bmatrix} \end{bmatrix} \\ \text{CONTENT} & \begin{bmatrix} \textit{put-relation} & \\ \text{PUTTER} & \boxed{1} \\ \text{THING-PUT} & \boxed{2} \\ \text{DESTINATION} & \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$


S
SYNSEM|LOC|CAT
HEAD 7
VAL
SUBJ ⟨⟩
COMPS ⟨⟩
SUBJ-DTR
HEAD-DTR
NP
PHON <John>
SYNSEM 4
VP
SYNSEM|LOC|CAT
HEAD 7
VAL
SUBJ ⟨4 NP⟩
COMPS ⟨⟩
HEAD-DTR
COMP-DTRS
V
PHON <put>
S|L
CAT
HEAD 7
verb
AUX −
VAL
SUBJ ⟨4 NP 1⟩
COMPS ⟨5 NP 2, 6 PP 3⟩
CONT
put-relation
PUTTER 1
THING-PUT 2
DESTINATION 3
NP
PHON <a book>
SYNSEM 5
PP
PHON <on the table>
SYNSEM 6

The Chomsky Hierarchy



The Chomsky Hierarchy

Type	Common Name	Rule Skeleton	Linguistic Example
0	Turing Equivalent	$\alpha \to \beta$, s.t. $\alpha \neq \epsilon$	HPSG, LFG, Minimalism
1	Context Sensitive	$\alpha A \beta \to \alpha \gamma \beta$, s.t. $\gamma \neq \epsilon$	
–	Mildly Context Sensitive		TAG, CCG
2	Context Free	$A \to \gamma$	Phrase-Structure Grammars
3	Regular	$A \to xB$ or $A \to x$	Finite-State Automata

- Head-Driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1987, 1994)
- Lexical Functional Grammar (LFG) (Bresnan, 1982)
- Minimalist Grammar (Stabler, 1997)

- Tree-Adjoining Grammars (TAG) (Joshi, 1985)
- Combinatory Categorial Grammars (CCG) (Steedman, 1996, 2000)

Turing Test

- **Turing Test**: Interrogator ‘c’ engages in a natural language conversation with ‘a’ and ‘b’ to determine which is a computer and which is a human.