

Ling 566

Oct 6, 2016

Feature Structures
Headed Rules, Trees

Overview

- Review: problems with CFG, modeling
- Feature structures, unification (pizza)
- Features for linguistic description
- Reformulate grammar rules
- Notion of head/headedness
- Licensing of trees
- Reading questions

Our Goals

- Descriptive, generative grammar
 - Describing English (in this case)
 - Generating all possible well-formed sentences (and no ill-formed ones)
 - Assigning appropriate structures
- Design/discover an appropriate *type* of model (through incremental improvement)
- Create a particular model (grammar fragment) for English

Problems with Context-Free Grammar (atomic node labels)

- Potentially arbitrary rules
- Gets clunky quickly with cross-cutting properties
- Not quite powerful enough for natural languages

Solution: Replace atomic node labels with feature structures.

Cross-cutting Grammatical Properties

3rd singular subject

plural subject

direct object NP

denies

deny

no direct object NP

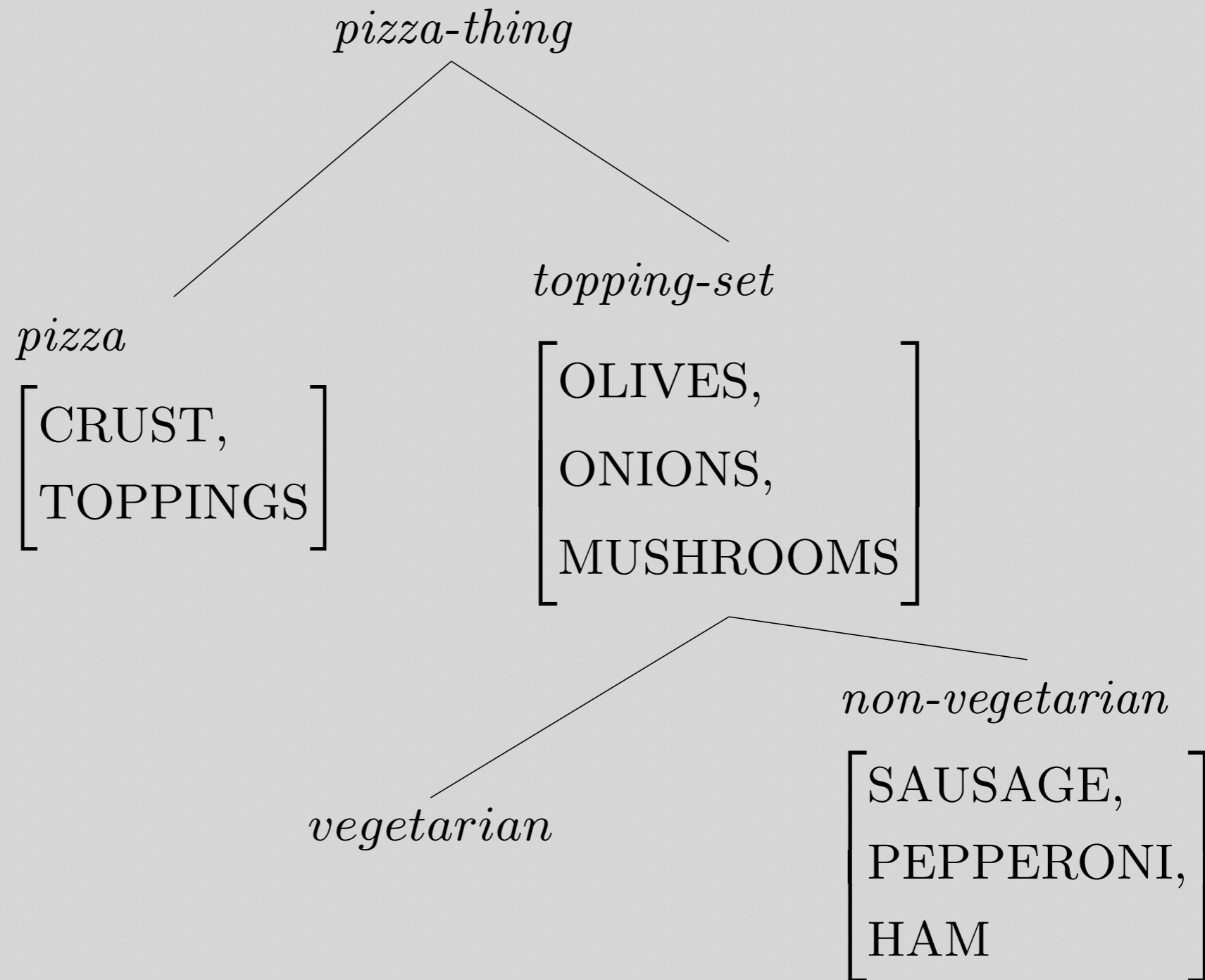
disappears

disappear

Feature Structure Descriptions

FEATURE ₁	VALUE ₁
FEATURE ₂	VALUE ₂
...	
FEATURE _n	VALUE _n

A Pizza Type Hierarchy



TYPE	FEATURES/VALUES	IST
<i>pizza-thing</i>		
<i>pizza</i>	$\left[\begin{array}{ll} \text{CRUST} & \{ \text{thick, thin, stuffed} \} \\ \text{TOPPINGS} & \text{topping-set} \end{array} \right]$	<i>pizza-thing</i>
<i>topping-set</i>	$\left[\begin{array}{ll} \text{OLIVES} & \{ +, - \} \\ \text{ONIONS} & \{ +, - \} \\ \text{MUSHROOMS} & \{ +, - \} \end{array} \right]$	<i>pizza-thing</i>
<i>vegetarian</i>		<i>topping-set</i>
<i>non-vegetarian</i>	$\left[\begin{array}{ll} \text{SAUSAGE} & \{ +, - \} \\ \text{PEPPERONI} & \{ +, - \} \\ \text{HAM} & \{ +, - \} \end{array} \right]$	<i>topping-set</i>

Type Hierarchies

A type hierarchy....

- ... states what kinds of objects we claim exist (the types)
- ... organizes the objects hierarchically into classes with shared properties (the type hierarchy)
- ... states what general properties each kind of object has (the feature and feature value declarations).

Pizza Descriptions and Pizza Models

$$\left[\begin{array}{l} \textit{pizza} \\ \text{CRUST} \quad \text{thick} \\ \text{TOPPINGS} \quad \left[\begin{array}{l} \textit{vegetarian} \\ \text{OLIVES} \quad + \\ \text{ONIONS} \quad + \end{array} \right] \end{array} \right]$$

How many pizza models (by definition, fully resolved) satisfy this description?

Answer: 2

$$\left[\begin{array}{l} \textit{pizza} \\ \text{CRUST} \quad \text{thick} \\ \text{TOPPINGS} \quad \left[\begin{array}{l} \textit{vegetarian} \\ \text{OLIVES} \quad + \\ \text{ONIONS} \quad + \end{array} \right] \end{array} \right]$$

{<CRUST , thick> , <TOPPINGS , { <OLIVES , + > , <ONIONS, +> , <MUSHROOMS, ->}>}

{<CRUST , thick> , <TOPPINGS , { <OLIVES , + > , <ONIONS, +> , <MUSHROOMS, +>}>}

Pizza Descriptions and Pizza Models

$$\left[\begin{array}{l} \textit{pizza} \\ \text{CRUST} \quad \text{thick} \\ \text{TOPPINGS} \quad \left[\begin{array}{l} \textit{vegetarian} \\ \text{OLIVES} \quad + \\ \text{ONIONS} \quad + \end{array} \right] \end{array} \right]$$

How many pizzas-in-the-world do the pizza models correspond to?

Answer: A large, constantly-changing number.

Pizza Descriptions and Pizza Models

$$\left[\begin{array}{l} \textit{pizza} \\ \text{CRUST} \quad \text{thick} \\ \text{TOPPINGS} \quad \left[\begin{array}{l} \textit{vegetarian} \\ \text{OLIVES} \quad + \\ \text{ONIONS} \quad + \end{array} \right] \end{array} \right]$$

‘type’/‘token’ distinction
applies to sentences as well

Combining Constraints

$$\left[\begin{array}{l} \textit{pizza} \\ \text{CRUST} \\ \text{TOPPINGS} \end{array} \right] \begin{array}{l} \text{thick} \\ \left[\begin{array}{l} \text{OLIVES} \\ \text{HAM} \end{array} \right] \begin{array}{l} + \\ - \end{array} \end{array} \right] \& \left[\begin{array}{l} \textit{pizza} \\ \text{TOPPINGS} \end{array} \right] \left[\begin{array}{l} \text{OLIVES} \\ \text{ONIONS} \end{array} \right] \begin{array}{l} + \\ + \end{array} \right]$$

Combining Constraints

<i>pizza</i>	
CRUST	thick
	[OLIVES +]
TOPPINGS	[ONIONS +]
	[HAM -]

Combining Constraints

$$\left[\begin{array}{l} \text{pizza} \\ \text{CRUST} \\ \text{TOPPINGS} \end{array} \begin{array}{l} \text{thick} \\ \left[\begin{array}{l} \text{OLIVES} \\ \text{HAM} \end{array} \right] \begin{array}{l} + \\ - \end{array} \end{array} \right] \& \left[\begin{array}{l} \text{pizza} \\ \text{CRUST} \\ \text{TOPPINGS} \end{array} \begin{array}{l} \text{thin} \\ \left[\begin{array}{l} \text{OLIVES} \\ \text{ONIONS} \end{array} \right] \begin{array}{l} + \\ + \end{array} \end{array} \right] \\ = \emptyset$$

Combining Constraints

$$\left[\begin{array}{l} \textit{pizza} \\ \text{CRUST} \\ \text{TOPPINGS} \end{array} \begin{array}{l} \text{thick} \\ \left[\begin{array}{l} \text{OLIVES} \\ \text{HAM} \end{array} \right] \begin{array}{l} + \\ + \end{array} \end{array} \right] \& \left[\begin{array}{l} \textit{pizza} \\ \text{CRUST} \\ \text{TOPPINGS} \end{array} \begin{array}{l} \text{thick} \\ \textit{vegetarian} \end{array} \right]$$

$$= \emptyset$$

Combining Constraints

$$\left[\begin{array}{l} \textit{pizza} \\ \text{CRUST} \\ \text{TOPPINGS} \end{array} \begin{array}{l} \text{thick} \\ \left[\begin{array}{l} \text{OLIVES} \\ \text{HAM} \end{array} \right] \begin{array}{l} + \\ - \end{array} \end{array} \right] \& \left[\begin{array}{l} \textit{pizza} \\ \text{CRUST} \\ \text{TOPPINGS} \end{array} \begin{array}{l} \text{thick} \\ \textit{vegetarian} \end{array} \right]$$

$$= \emptyset$$

A New Theory of Pizzas

pizza : $\left[\begin{array}{l} \text{CRUST} \quad \left\{ \text{thick , thin , stuffed} \right\} \\ \text{ONE-HALF} \quad \textit{topping-set} \\ \text{OTHER-HALF} \quad \textit{topping-set} \end{array} \right]$

Combining Constraints

$$\begin{array}{l} \left[\begin{array}{l} \textit{pizza} \\ \text{ONE-HALF} \end{array} \right. \left[\begin{array}{l} \text{ONIONS} \quad + \\ \text{OLIVES} \quad - \end{array} \right] \end{array} \quad \& \quad \begin{array}{l} \left[\begin{array}{l} \textit{pizza} \\ \text{OTHER-HALF} \end{array} \right. \left[\begin{array}{l} \text{ONIONS} \quad - \\ \text{OLIVES} \quad + \end{array} \right] \end{array} \\ \\ = \\ \begin{array}{l} \left[\begin{array}{l} \textit{pizza} \\ \text{ONE-HALF} \\ \text{OTHER-HALF} \end{array} \right. \left[\begin{array}{l} \text{ONIONS} \quad + \\ \text{OLIVES} \quad - \end{array} \right] \\ \left[\begin{array}{l} \text{ONIONS} \quad - \\ \text{OLIVES} \quad + \end{array} \right] \end{array} \end{array}$$

Identity Constraints (tags)

<i>pizza</i>					
CRUST	thin				
ONE-HALF	<table><tr><td>OLIVES</td><td>1</td></tr><tr><td>ONIONS</td><td>2</td></tr></table>	OLIVES	1	ONIONS	2
OLIVES	1				
ONIONS	2				
OTHER-HALF	<table><tr><td>OLIVES</td><td>1</td></tr><tr><td>ONIONS</td><td>2</td></tr></table>	OLIVES	1	ONIONS	2
OLIVES	1				
ONIONS	2				

Combining Constraints

$$\left[\begin{array}{l} \textit{pizza} \\ \text{ONE-HALF} \\ \text{OTHER-HALF} \end{array} \begin{array}{l} \boxed{1} \\ \boxed{1} \end{array} \left[\begin{array}{l} \text{ONIONS} \\ \text{OLIVES} \end{array} \begin{array}{l} + \\ - \end{array} \right] \& \left[\begin{array}{l} \textit{pizza} \\ \text{OTHER-HALF} \end{array} \left[\begin{array}{l} \text{MUSHROOMS} \\ \text{OLIVES} \end{array} \begin{array}{l} - \\ - \end{array} \right] \right]$$

=

$$\left[\begin{array}{l} \textit{pizza} \\ \text{ONE-HALF} \\ \text{OTHER-HALF} \end{array} \begin{array}{l} \\ \boxed{1} \\ \boxed{1} \end{array} \left[\begin{array}{l} \text{ONIONS} \\ \text{OLIVES} \\ \text{MUSHROOMS} \end{array} \begin{array}{l} + \\ - \\ - \end{array} \right] \right]$$

Note

$$\left[\begin{array}{l} \textit{pizza} \\ \\ \text{ONE-HALF} \\ \\ \text{OTHER-HALF} \end{array} \right] \begin{array}{l} \\ \boxed{1} \\ \\ \boxed{1} \end{array} \left[\begin{array}{l} \text{ONIONS} \\ \text{OLIVES} \\ \text{MUSHROOMS} \end{array} \right] \begin{array}{l} + \\ - \\ - \end{array}$$

=

$$\left[\begin{array}{l} \textit{pizza} \\ \text{ONE-HALF} \\ \\ \text{OTHER-HALF} \end{array} \right] \begin{array}{l} \boxed{1} \\ \\ \boxed{1} \end{array} \left[\begin{array}{l} \text{ONIONS} \\ \text{OLIVES} \\ \text{MUSHROOMS} \end{array} \right] \begin{array}{l} + \\ - \\ - \end{array}$$

Combining Constraints

$$\left[\begin{array}{l} \textit{pizza} \\ \text{ONE-HALF} \\ \text{OTHER-HALF} \end{array} \right] \left[\begin{array}{l} \boxed{1} \left[\begin{array}{l} \text{ONIONS} \\ \text{OLIVES} \end{array} \right] \\ \boxed{1} \textit{vegetarian} \end{array} \right] \begin{array}{l} + \\ + \end{array} \right] \& \left[\begin{array}{l} \textit{pizza} \\ \text{ONE-HALF} \end{array} \right] \left[\begin{array}{l} \text{SAUSAGE} \\ \text{HAM} \end{array} \right] \begin{array}{l} + \\ - \end{array} \right]$$

$$= \emptyset$$

Why combine constraints?

- The pizza example illustrates how unification can be used to combine information from different sources.
- In our grammar, information will come from lexical entries, grammar rules, and general principles.

Linguistic Application of Feature Structures: Making the Mnemonic Meaningful

What do these CFG categories have in common?

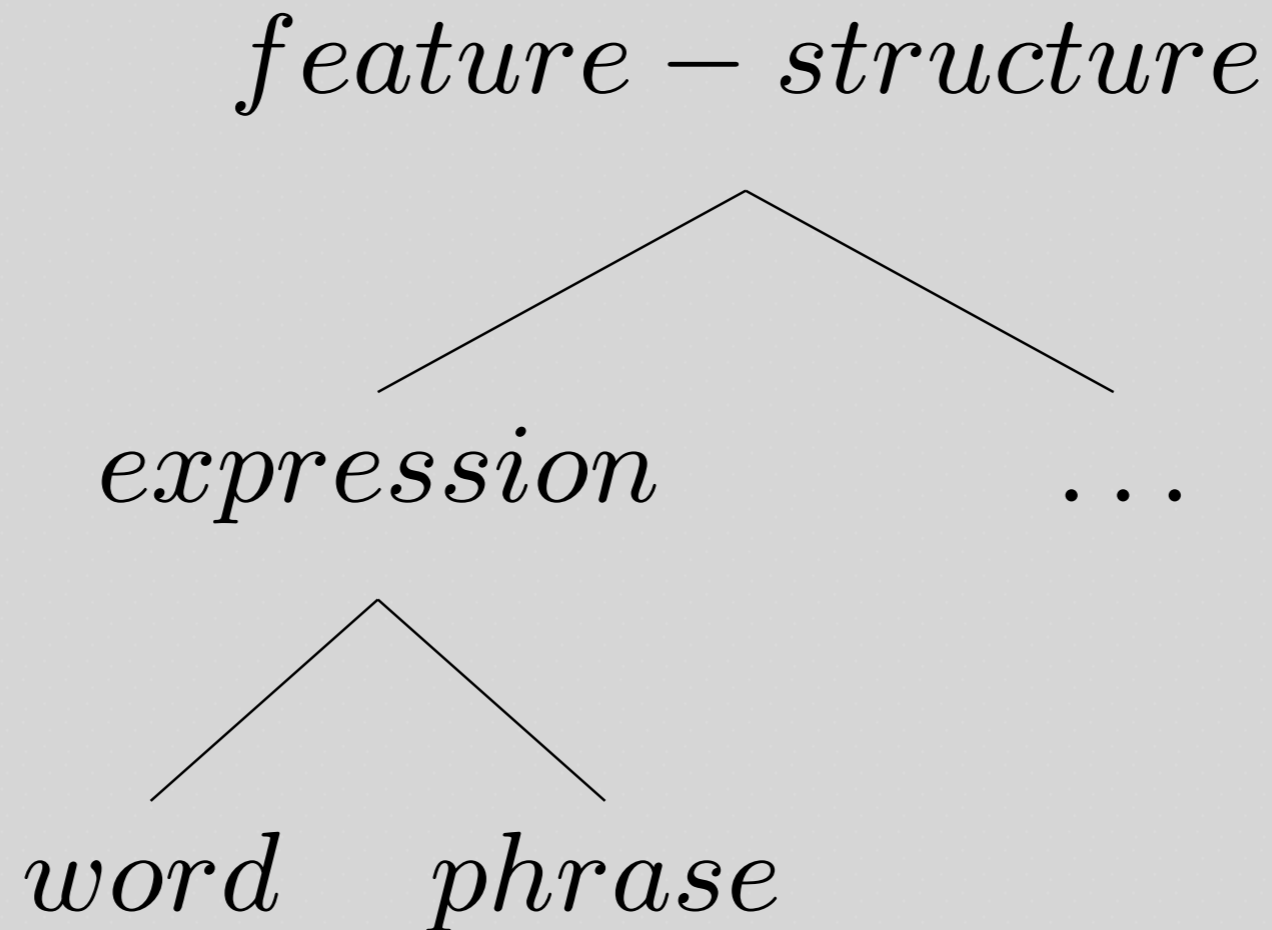
NP & VP: are both phrases

N & V: are both words

NP & N: are both 'nouny'

VP & V: are both 'verby'

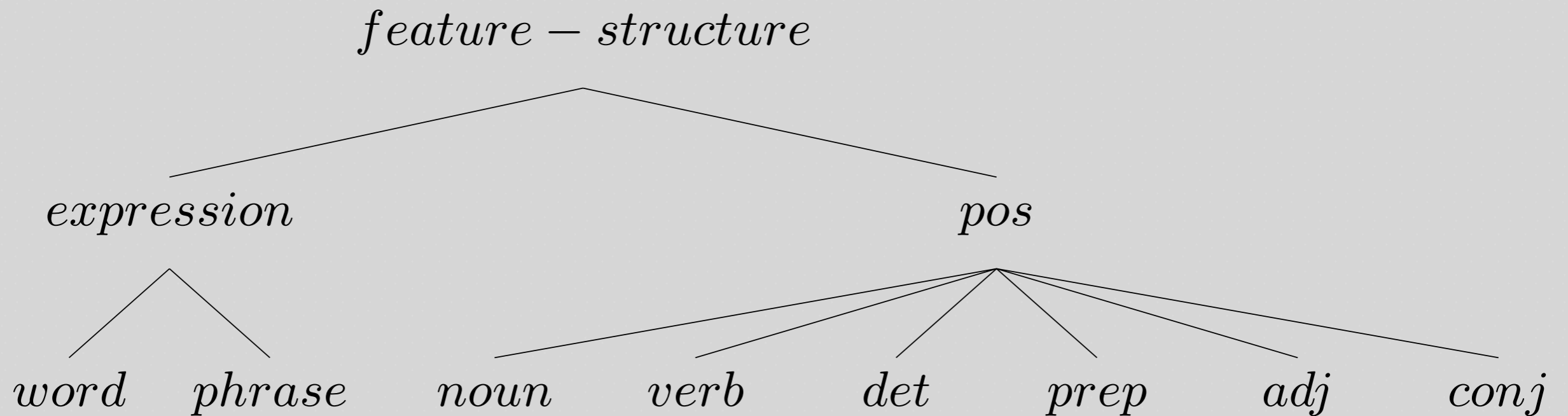
The Beginnings of Our Type Hierarchy



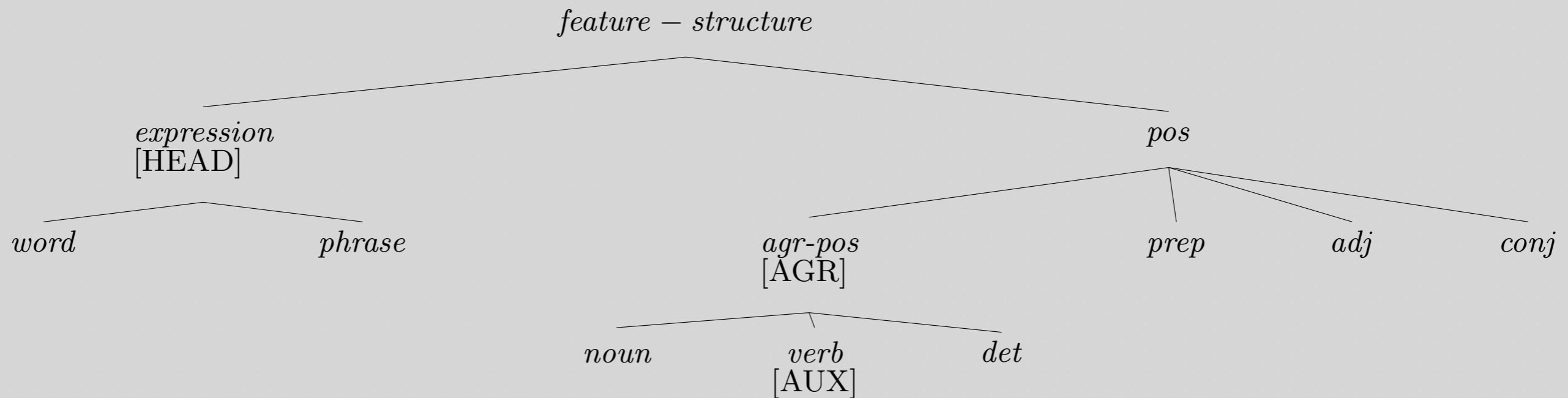
A Feature for Part of Speech

$$\text{NP} = \left[\begin{array}{l} \textit{phrase} \\ \text{HEAD} \quad \textit{noun} \end{array} \right]$$
$$\left\langle \text{bird} , \left[\begin{array}{l} \textit{word} \\ \text{HEAD} \quad \textit{noun} \end{array} \right] \right\rangle$$

Type Hierarchy for Parts of Speech I



Type Hierarchy for Parts of Speech II



A Feature for Valence

$$IV = \begin{bmatrix} \textit{word} \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & [\text{COMPS} \quad \textit{itr}] \end{bmatrix}$$

$$TV = \begin{bmatrix} \textit{word} \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & [\text{COMPS} \quad \textit{str}] \end{bmatrix}$$

$$DTV = \begin{bmatrix} \textit{word} \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & [\text{COMPS} \quad \textit{dtr}] \end{bmatrix}$$

Underspecification

$$V = \begin{bmatrix} \textit{word} \\ \text{HEAD} \quad \textit{verb} \end{bmatrix}$$

$$VP = \begin{bmatrix} \textit{phrase} \\ \text{HEAD} \quad \textit{verb} \end{bmatrix}$$

$$[\text{HEAD} \quad \textit{verb}]$$

Another Valence Feature

$$\text{NP} = \begin{bmatrix} \textit{phrase} \\ \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \textit{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix}$$

$$\text{NOM} = \begin{bmatrix} \textit{phrase} \\ \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \textit{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

SPR and Verbs

$$S = \left[\begin{array}{l} \textit{phrase} \\ \text{HEAD} \quad \textit{verb} \\ \text{VAL} \quad \left[\begin{array}{l} \text{COMPS} \quad \textit{itr} \\ \text{SPR} \quad + \end{array} \right] \end{array} \right]$$

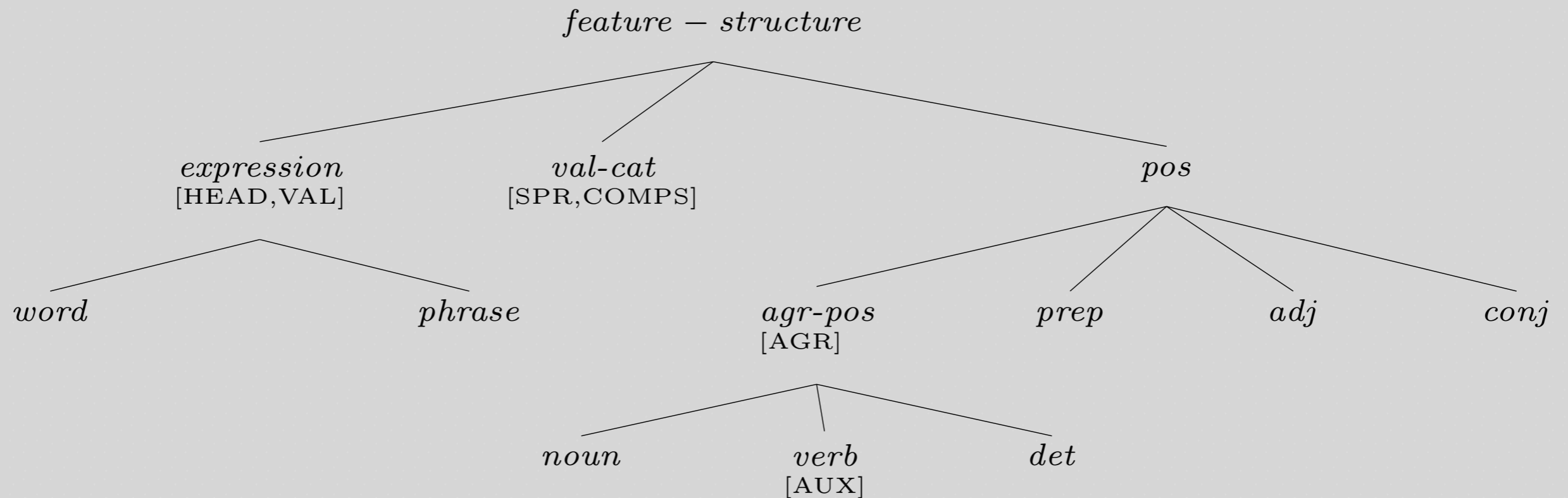
$$VP = \left[\begin{array}{l} \textit{phrase} \\ \text{HEAD} \quad \textit{verb} \\ \text{VAL} \quad \left[\begin{array}{l} \text{COMPS} \quad \textit{itr} \\ \text{SPR} \quad - \end{array} \right] \end{array} \right]$$

S and NP

$$\left[\text{VAL} \left[\begin{array}{l} \text{COMPS} \quad \text{itr} \\ \text{SPR} \quad \quad + \end{array} \right] \right]$$

- We created a monster
- our creation of a monster

Type Hierarchy So Far



Reformulating the Grammar Rules I

Which Ch 2 rules do these correspond to?

Head-Complement Rule 1:

$$\left[\begin{array}{l} \textit{phrase} \\ \text{VAL} \left[\begin{array}{l} \text{COMPS itr} \\ \text{SPR} - \end{array} \right] \end{array} \right] \rightarrow \mathbf{H} \left[\begin{array}{l} \textit{word} \\ \text{VAL} \left[\begin{array}{l} \text{COMPS itr} \\ \text{SPR} - \end{array} \right] \end{array} \right]$$

Head Complement Rule 2:

$$\left[\begin{array}{l} \textit{phrase} \\ \text{VAL} \left[\begin{array}{l} \text{COMPS itr} \\ \text{SPR} - \end{array} \right] \end{array} \right] \rightarrow \mathbf{H} \left[\begin{array}{l} \textit{word} \\ \text{VAL} \left[\begin{array}{l} \text{COMPS str} \\ \text{SPR} - \end{array} \right] \end{array} \right] \text{NP}$$

Head Complement Rule 3:

$$\left[\begin{array}{l} \textit{phrase} \\ \text{VAL} \left[\begin{array}{l} \text{COMPS itr} \\ \text{SPR} - \end{array} \right] \end{array} \right] \rightarrow \mathbf{H} \left[\begin{array}{l} \textit{word} \\ \text{VAL} \left[\begin{array}{l} \text{COMPS dtr} \\ \text{SPR} - \end{array} \right] \end{array} \right] \text{NP NP}$$

Reformulating the Grammar Rules II

Head-Specifier Rule 1:

$$\left[\begin{array}{l} \textit{phrase} \\ \text{VAL} \left[\begin{array}{l} \text{COMPS} \quad \textit{itr} \\ \text{SPR} \quad + \end{array} \right] \end{array} \right] \rightarrow \left[\begin{array}{l} \text{NP} \\ \text{HEAD} \left[\begin{array}{l} \text{AGR} \quad \boxed{1} \end{array} \right] \end{array} \right] \mathbf{H} \left[\begin{array}{l} \textit{phrase} \\ \text{HEAD} \left[\begin{array}{l} \textit{verb} \\ \text{AGR} \quad \boxed{1} \end{array} \right] \\ \text{VAL} \left[\begin{array}{l} \text{SPR} \quad - \end{array} \right] \end{array} \right]$$

Head-Specifier Rule 2:

$$\left[\begin{array}{l} \textit{phrase} \\ \text{VAL} \left[\begin{array}{l} \text{COMPS} \quad \textit{itr} \\ \text{SPR} \quad + \end{array} \right] \end{array} \right] \rightarrow \text{D} \mathbf{H} \left[\begin{array}{l} \textit{phrase} \\ \text{HEAD} \quad \textit{noun} \\ \text{VAL} \left[\begin{array}{l} \text{SPR} \quad - \end{array} \right] \end{array} \right]$$

Reformulating the Grammar Rules III

Non-Branching NP Rule

$$\left[\begin{array}{l} \textit{phrase} \\ \text{VAL} \left[\begin{array}{l} \text{COMPS} \textit{ itr} \\ \text{SPR} \textit{ +} \end{array} \right] \end{array} \right] \rightarrow \mathbf{H} \left[\begin{array}{l} \textit{word} \\ \text{HEAD} \textit{ noun} \\ \text{VAL} \left[\begin{array}{l} \text{SPR} \textit{ +} \end{array} \right] \end{array} \right]$$

Head-Modifier Rule

$$\left[\begin{array}{l} \textit{phrase} \\ \text{VAL} \left[\begin{array}{l} \text{COMPS} \textit{ itr} \\ \text{SPR} \textit{ -} \end{array} \right] \end{array} \right] \rightarrow \mathbf{H} \left[\begin{array}{l} \textit{phrase} \\ \text{VAL} \left[\begin{array}{l} \text{SPR} \textit{ -} \end{array} \right] \end{array} \right] \text{PP}$$

Coordination Rule

$$\boxed{1} \rightarrow \boxed{1}^+ \left[\begin{array}{l} \textit{word} \\ \text{HEAD} \textit{ conj} \end{array} \right] \boxed{1}$$

Advantages of the New Formulation

- Subject-verb agreement is stipulated only once (where?)
- Common properties of verbs with different valences are expressed by common features (for example?)
- Parallelisms across phrase types are captured (for example?)

Disadvantages of the New Formulation

- We still have three head complement rules
- We still have two head specifier rules
- We only deal with three verb valences
(Which ones? What are some others?)
- The non-branching rule doesn't really do any empirical work
- Others?

Heads

- Intuitive idea: A phrase typically contains a word that determines its most essential properties, including
 - where it occurs in larger phrases, and
 - what its internal structure is
- This is called the head
- The term “head” is used both for the head word in a phrase and for all the intermediate phrases containing that word
- NB: Not all phrases have heads

Formalizing the Notion of Head

- Expressions have a feature HEAD
- HEAD's values are of type *pos*
- For HEAD values of type *agr-cat*, HEAD's value also includes the feature AGR
- Well-formed trees are subject to the Head Feature Principle

The Head Feature Principle

- Intuitive idea: Key properties of phrases are shared with their heads
- The HFP: In any headed phrase, the HEAD value of the mother and the head daughter must be identical.
- Sometimes described in terms of properties “percolating up” or “filtering down”, but this is just metaphorical talk

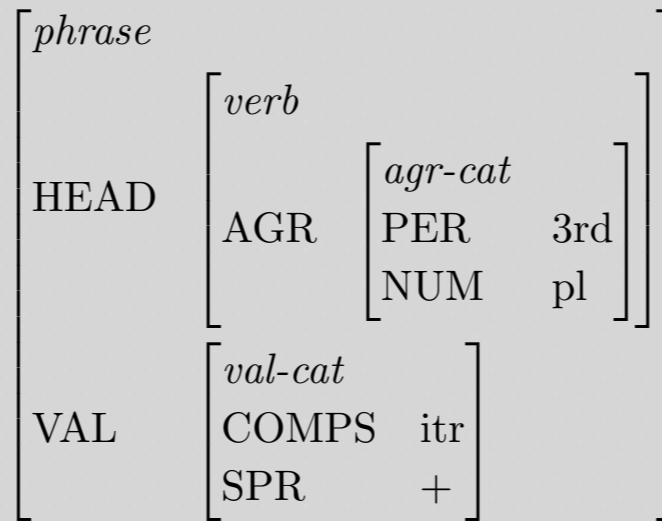
A Tree is Well-Formed if ...

- It and each subtree are licensed by a grammar rule or lexical entry
- All general principles (like the HFP) are satisfied.
- NB: Trees are part of our model of the language, so all their features have values (even though we will often be lazy and leave out the values irrelevant to our current point).

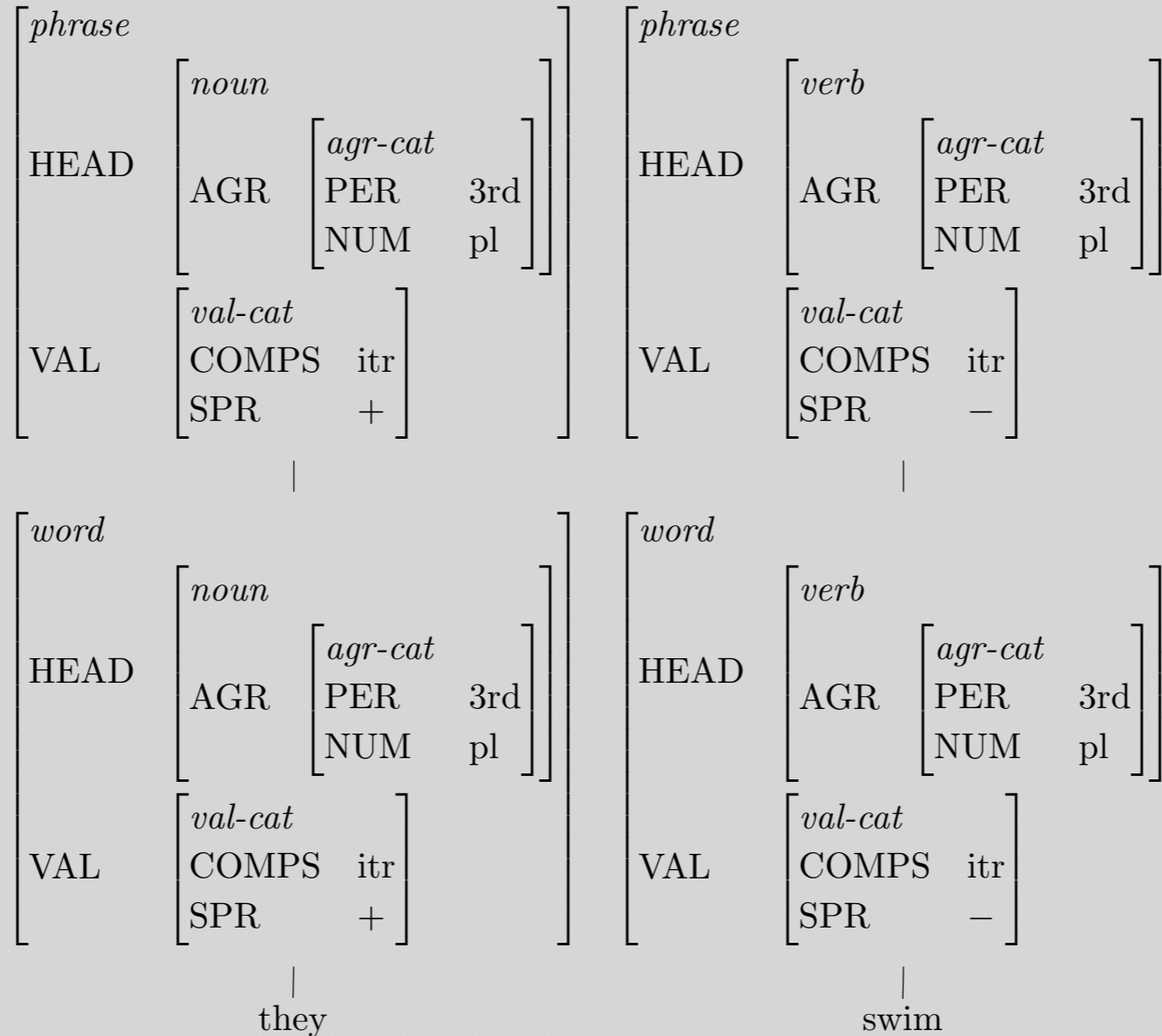
Question:

Do phrases that are not headed have
HEAD features?

Which rule licenses each node?

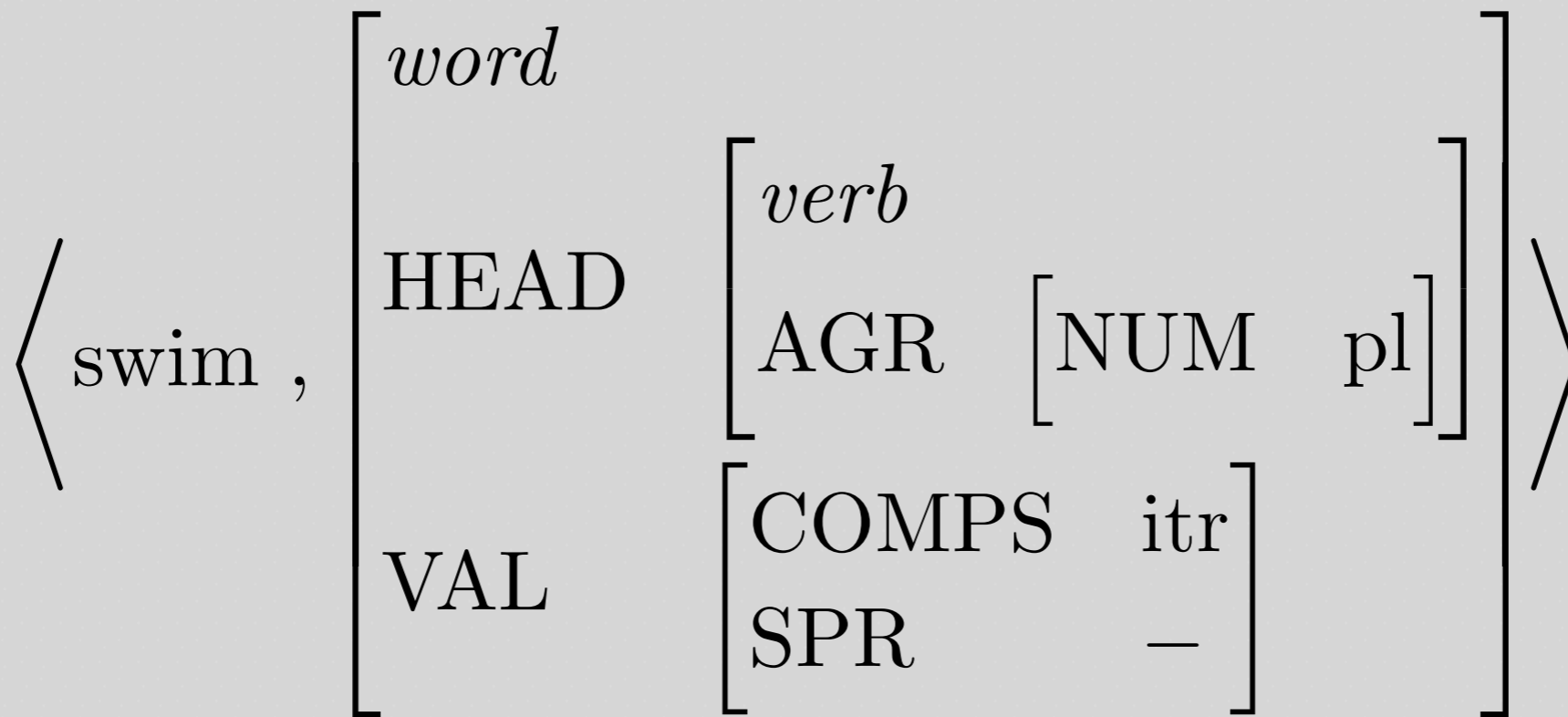


Note the three separate uses of DAGs



A Question:

Since the lexical entry for swim below has only [NUM pl] as the value of AGR, how did the tree on the previous slide get [PER 3rd] in the AGR of swim?



Overview

- Review: problems with CFG
- Modeling
- Feature structures, unification (pizza)
- Features for linguistic description
- Reformulate grammar rules
- Notion of head/headedness
- Licensing of trees
- Next time: Valence and agreement

Reading Questions

- Why are angle brackets used for both list-valued features and for lexical entries?
- How does the Ch 3 grammar differ from CFG?
- Can this kind of grammar handle naturally occurring (spoken) language?
- Should we be thinking in terms of [phrase HEAD verb [VAL [COMPS itr SPR +]]] and then converting that to VP when we need to abbreviate, or thinking in terms of VP and then switching when we need to delve in in more depth?

Reading Questions

- Is the choice of head daughter of a rule (where to put the H) arbitrary? Are there heuristics to decide?
- Can two daughters ever match the HEAD value on the mother?
- Why are phrases allowed to head other phrases? Why make VP head of S? Can we still talk about lexical items as the ultimate heads of things?
- Will the HFP do more for us than just determining POS?

Reading Questions

- What's up with SPR +/-?
- Is SPR similar to the notion of specifier in X-bar theory?
- Why are determiners [COMPS itr]? Don't Ds require elements to their right?
- How do we tell what value of SPR to use on various constituents in a tree?

Reading Questions

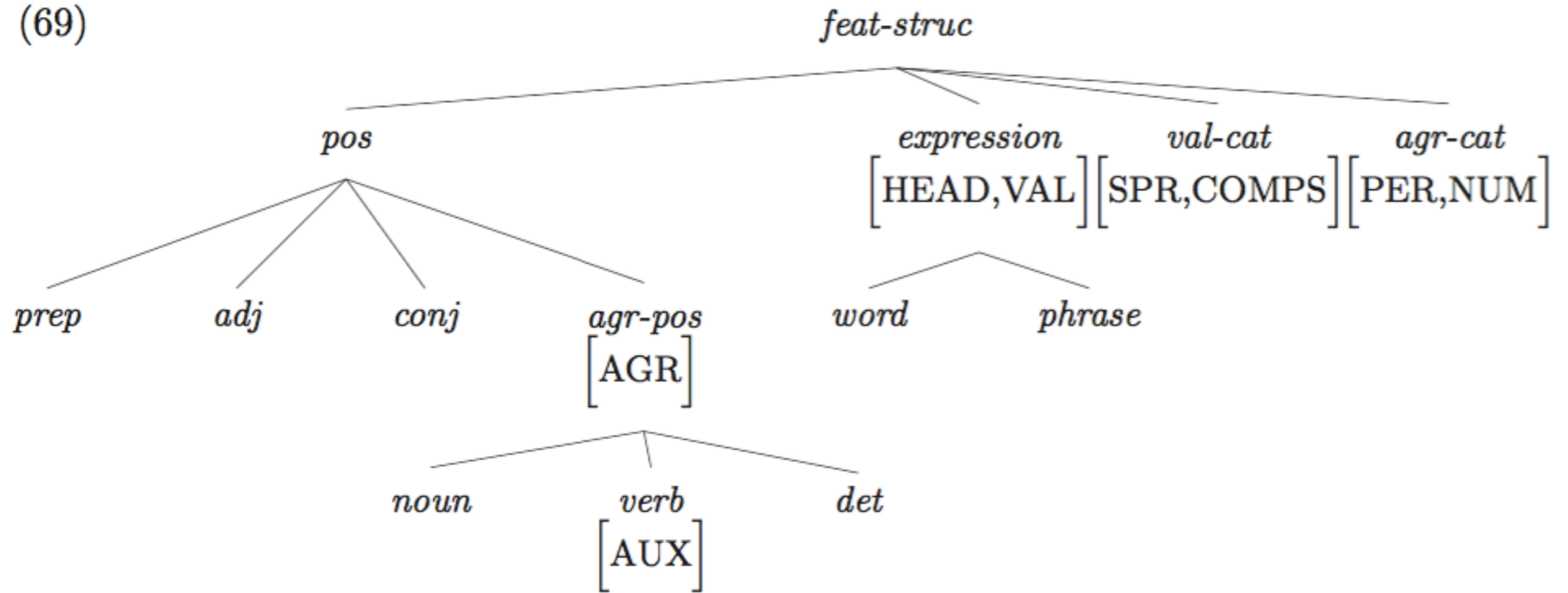
- What's the benefit of putting VAL features on all categories?
- What's the benefit of grouping SPR and COMPS under VAL?
- When/how do we check that the "need" for a COMPS/SPR is satisfied.

Reading Questions

- Why is HEAD on *expression* in (69), rather than on *pos*? (see fig next slide)
- What in the grammar tells us that HEAD takes something of type *pos*? (69) doesn't show this.
- Why is *word* under *expression* in the type hierarchy and not under *phrase*?

3.6.1 The Type Hierarchy

(69)



(SWB 2003:84)

<i>conj</i>		<i>pos</i>
<i>expression</i>	$\left[\begin{array}{ll} \text{HEAD} & \textit{pos} \\ \text{VAL} & \textit{val-cat} \end{array} \right]$	<i>feat-struct</i>
<i>word</i>		<i>expression</i>
<i>phrase</i>		<i>expression</i>
<i>val-cat</i>		<i>feat-struct</i>

(SWB 2003:84)

Reading Questions

- What is the inventory of all features used in all languages?
- When shall we use complete (resolved) structure and when can we omit some of the features when representing phrase structure tree?

Reading Questions

- How does the linguistic definition of constraint differ from the programming definition of constraint?
- How do you determine the head noun of a name or location that's comprised of two or more words where no one word seems to function as the head? Would you favor the first or last word as the head?