

Ling 566

Oct 3, 2024

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	<i>denies</i>	<i>deny</i>
no direct object NP	<i>disappears</i>	<i>disappear</i>

Two Kinds of Language Models

- Speakers' internalized knowledge (their grammar)
- Set of sentences in the language
- (Both distinct from: Probability distribution over strings)

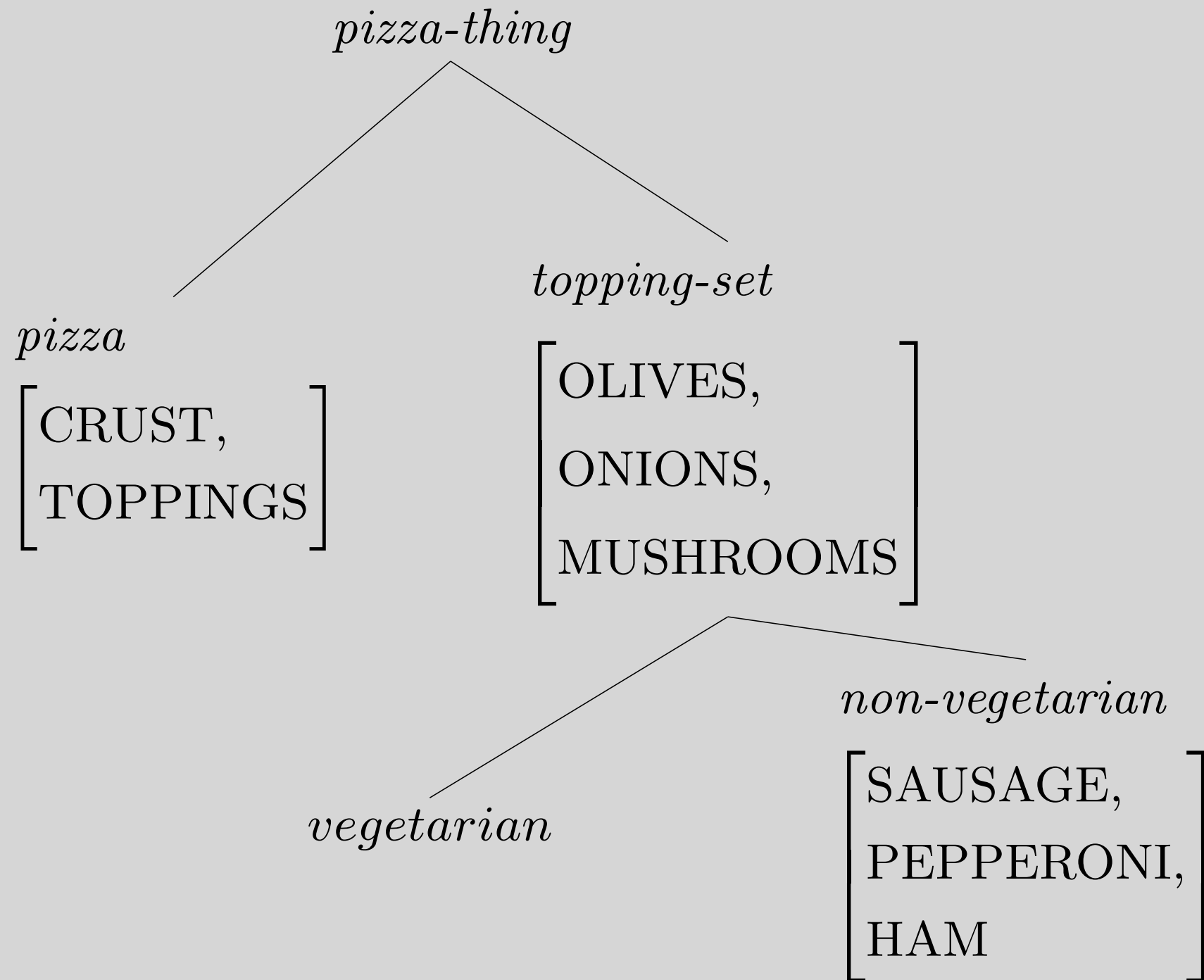
Things Involved in Modeling Language

- Real world entities (utterance types)
- Models (here: fully specified trees)
- Descriptions of the models (here: rules, principles, lexical entries)

Feature Structure Descriptions

$$\begin{bmatrix} \text{FEATURE}_1 & \text{VALUE}_1 \\ \text{FEATURE}_2 & \text{VALUE}_2 \\ \dots & \\ \text{FEATURE}_n & \text{VALUE}_n \end{bmatrix}$$

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

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

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

Answer: 2

<i>pizza</i>							
CRUST	thick						
TOPPINGS	<table><tr><td><i>vegetarian</i></td><td></td></tr><tr><td>OLIVES</td><td>+</td></tr><tr><td>ONIONS</td><td>+</td></tr></table>	<i>vegetarian</i>		OLIVES	+	ONIONS	+
<i>vegetarian</i>							
OLIVES	+						
ONIONS	+						

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

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

Pizza Descriptions and Pizza Models

$$\left[\begin{array}{cc} \text{pizza} & \\ \text{CRUST} & \text{thick} \\ \text{TOPPINGS} & \left[\begin{array}{c} \text{vegetarian} \\ \text{OLIVES} + \\ \text{ONIONS} + \end{array} \right] \end{array} \right]$$

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

Answer: A large, constantly-changing number.

Pizza Descriptions and Pizza Models

<i>pizza</i>	
CRUST	thick
TOPPINGS	<i>vegetarian</i>
	OLIVES +
	ONIONS +

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

Combining Constraints

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

Combining Constraints

<i>pizza</i>							
CRUST	thick						
TOPPINGS	<table><tr><td>OLIVES</td><td>+</td></tr><tr><td>ONIONS</td><td>+</td></tr><tr><td>HAM</td><td>-</td></tr></table>	OLIVES	+	ONIONS	+	HAM	-
OLIVES	+						
ONIONS	+						
HAM	-						

Combining Constraints

$$\begin{bmatrix} \text{pizza} \\ \text{CRUST} \\ \text{TOPPINGS} \end{bmatrix} \begin{matrix} \text{thick} \\ \left[\begin{matrix} \text{OLIVES} \\ \text{HAM} \end{matrix} \right] \begin{matrix} + \\ - \end{matrix} \end{matrix} \quad \& \quad \begin{bmatrix} \text{pizza} \\ \text{CRUST} \\ \text{TOPPINGS} \end{bmatrix} \begin{matrix} \text{thin} \\ \left[\begin{matrix} \text{OLIVES} \\ \text{ONIONS} \end{matrix} \right] \begin{matrix} + \\ + \end{matrix} \end{matrix} \\
 = \quad \phi$$

Combining Constraints

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

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

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

Combining Constraints

$$\begin{bmatrix} \textit{pizza} \\ \text{ONE-HALF} \end{bmatrix} \begin{bmatrix} \text{ONIONS} & + \\ \text{OLIVES} & - \end{bmatrix} \& \begin{bmatrix} \textit{pizza} \\ \text{OTHER-HALF} \end{bmatrix} \begin{bmatrix} \text{ONIONS} & - \\ \text{OLIVES} & + \end{bmatrix} \\[1em] = \\[1em] \begin{bmatrix} \textit{pizza} \\ \text{ONE-HALF} \\ \text{OTHER-HALF} \end{bmatrix} \begin{bmatrix} \text{ONIONS} & + \\ \text{OLIVES} & - \\ \text{ONIONS} & - \\ \text{OLIVES} & + \end{bmatrix}$$

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

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

=

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

Note

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

=

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

Combining Constraints

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

$$= \emptyset$$

How badly do you want pizza now?



PIZZA! NOW!

0

I can wait until dinner

0

Meh

0

I've been eating pizza this whole time

0

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?

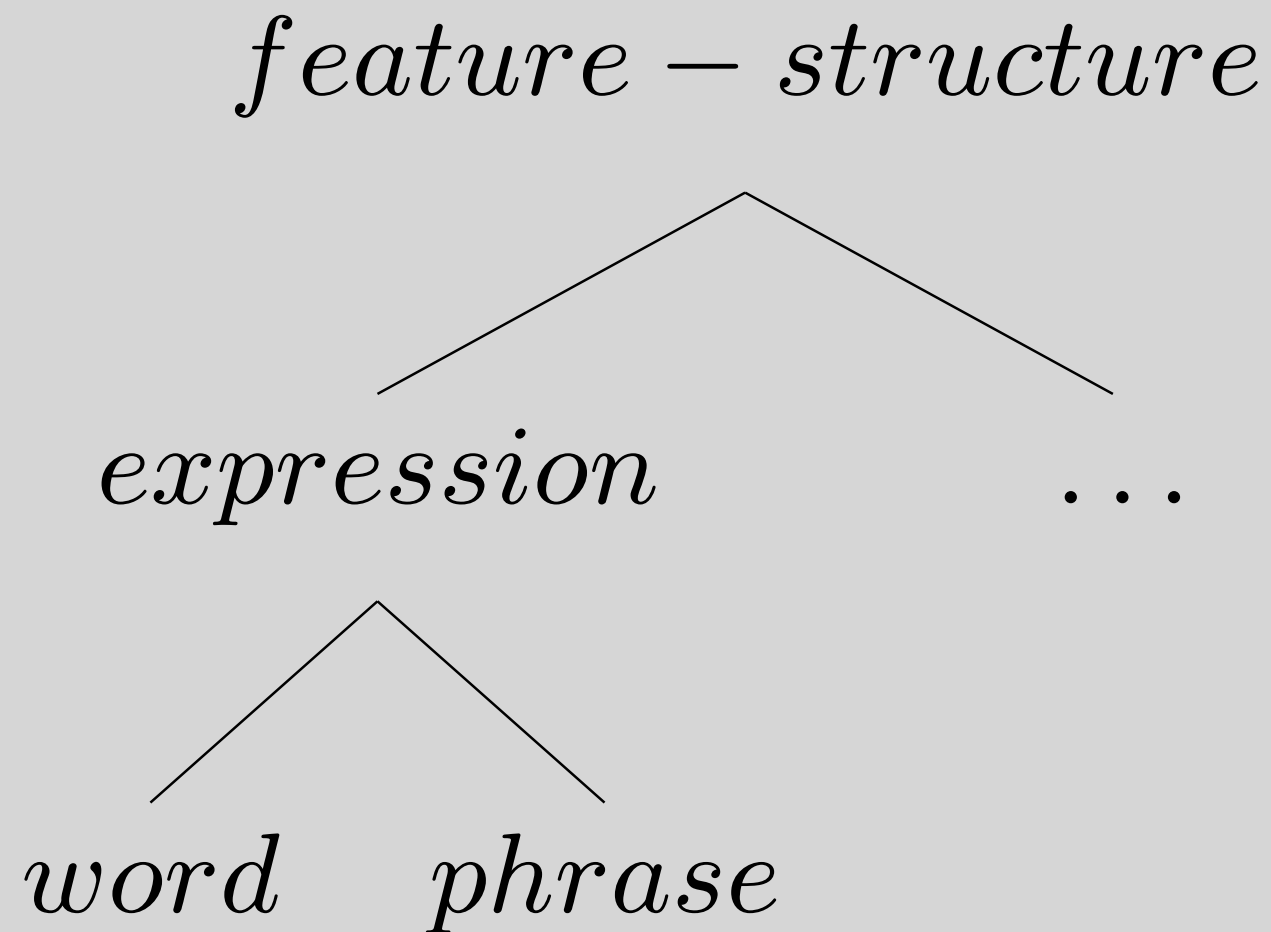
NP & VP:	are both phrases
----------	------------------

N & V:	are both words
--------	----------------

NP & N:	are both ‘nouny’
---------	------------------

VP & V:	are both ‘verby’
---------	------------------

The Beginnings of Our Type Hierarchy

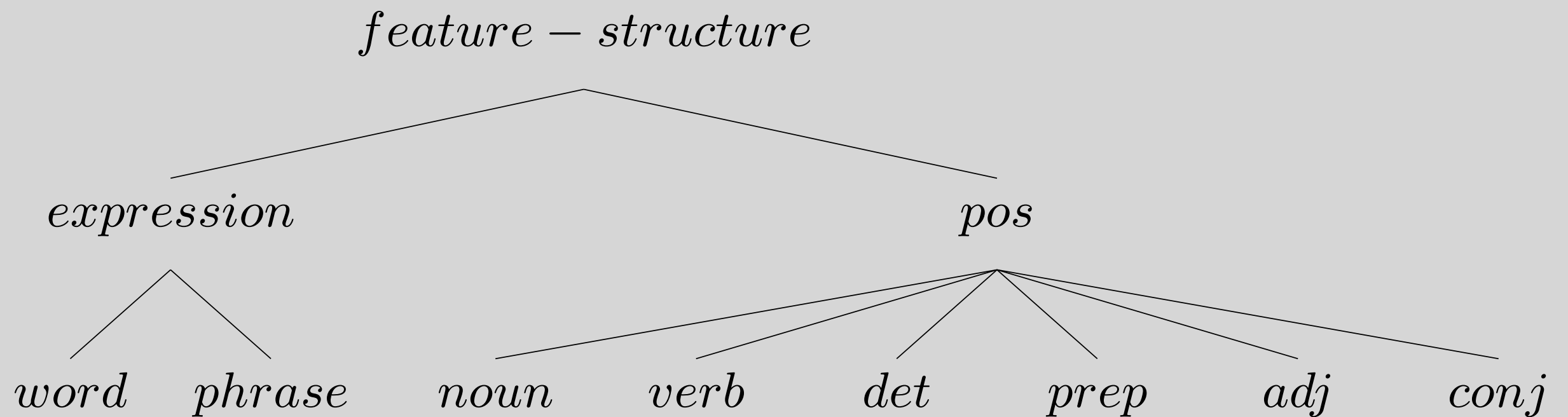


A Feature for Part of Speech

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

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

Type Hierarchy for Parts of Speech I



Having 'expression' and 'pos' at the same level of that hierarchy



Bugs me

0

Didn't really stand out to me at all

0

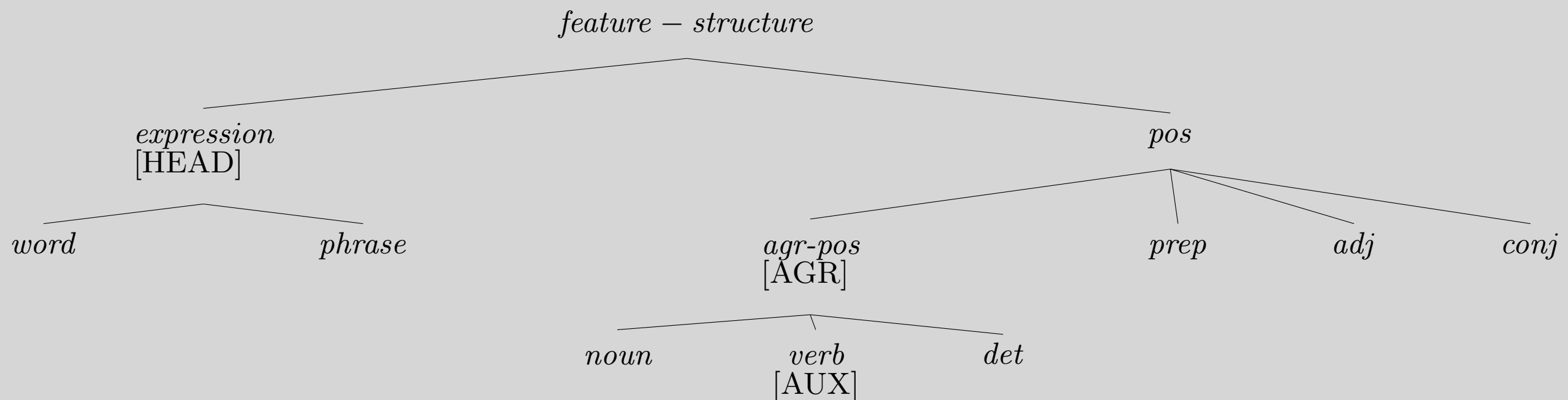
Makes sense

0

None of the above

0

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

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

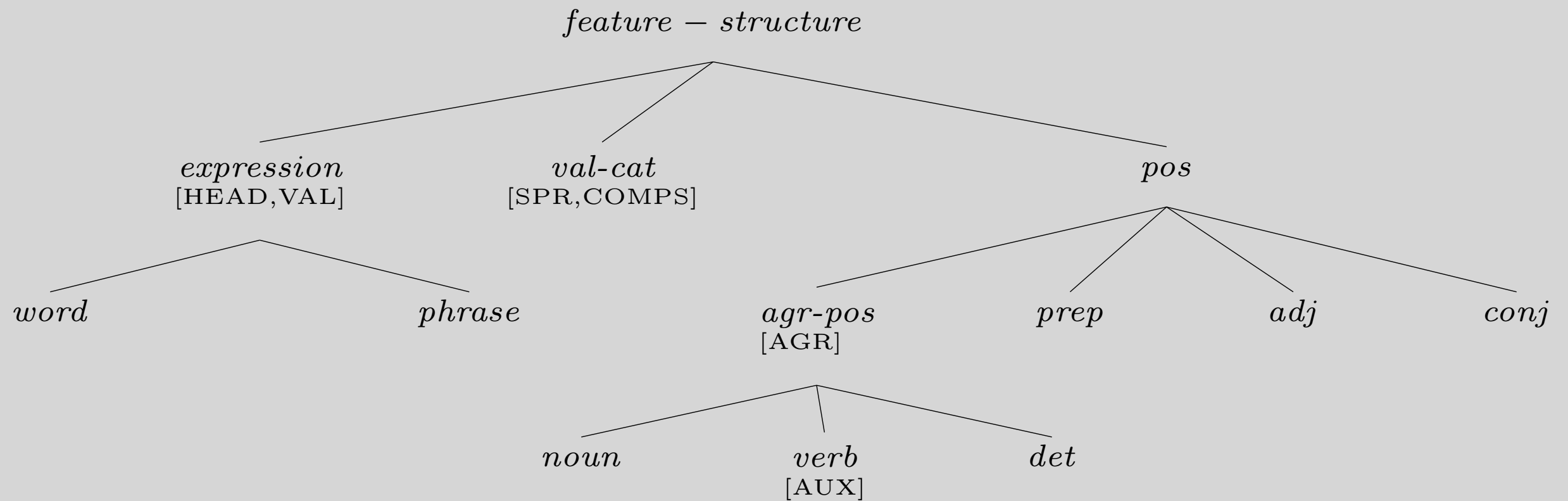
$$VP = \left[\begin{array}{cc} \textit{phrase} & \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & \left[\begin{array}{cc} \text{COMPS} & \textit{itr} \\ \text{SPR} & - \end{array} \right] \end{array} \right]$$

S and NP

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

- 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:

$$\begin{bmatrix} \textit{phrase} \\ \text{VAL} \begin{bmatrix} \text{COMPS} & \textit{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H} \begin{bmatrix} \textit{word} \\ \text{VAL} \begin{bmatrix} \text{COMPS} & \textit{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

Head Complement Rule 2:

$$\begin{bmatrix} \textit{phrase} \\ \text{VAL} \begin{bmatrix} \text{COMPS} & \textit{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H} \begin{bmatrix} \textit{word} \\ \text{VAL} \begin{bmatrix} \text{COMPS} & \textit{str} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \text{NP}$$

Head Complement Rule 3:

$$\begin{bmatrix} \textit{phrase} \\ \text{VAL} \begin{bmatrix} \text{COMPS} & \textit{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H} \begin{bmatrix} \textit{word} \\ \text{VAL} \begin{bmatrix} \text{COMPS} & \textit{dtr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \text{NP NP}$$

Reformulating the Grammar Rules II

Head-Specifier Rule 1:

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

Head-Specifier Rule 2:

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

Reformulating the Grammar Rules III

Non-Branching NP Rule

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

Head-Modifier Rule

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

Coordination Rule

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

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-pos*, 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 ...

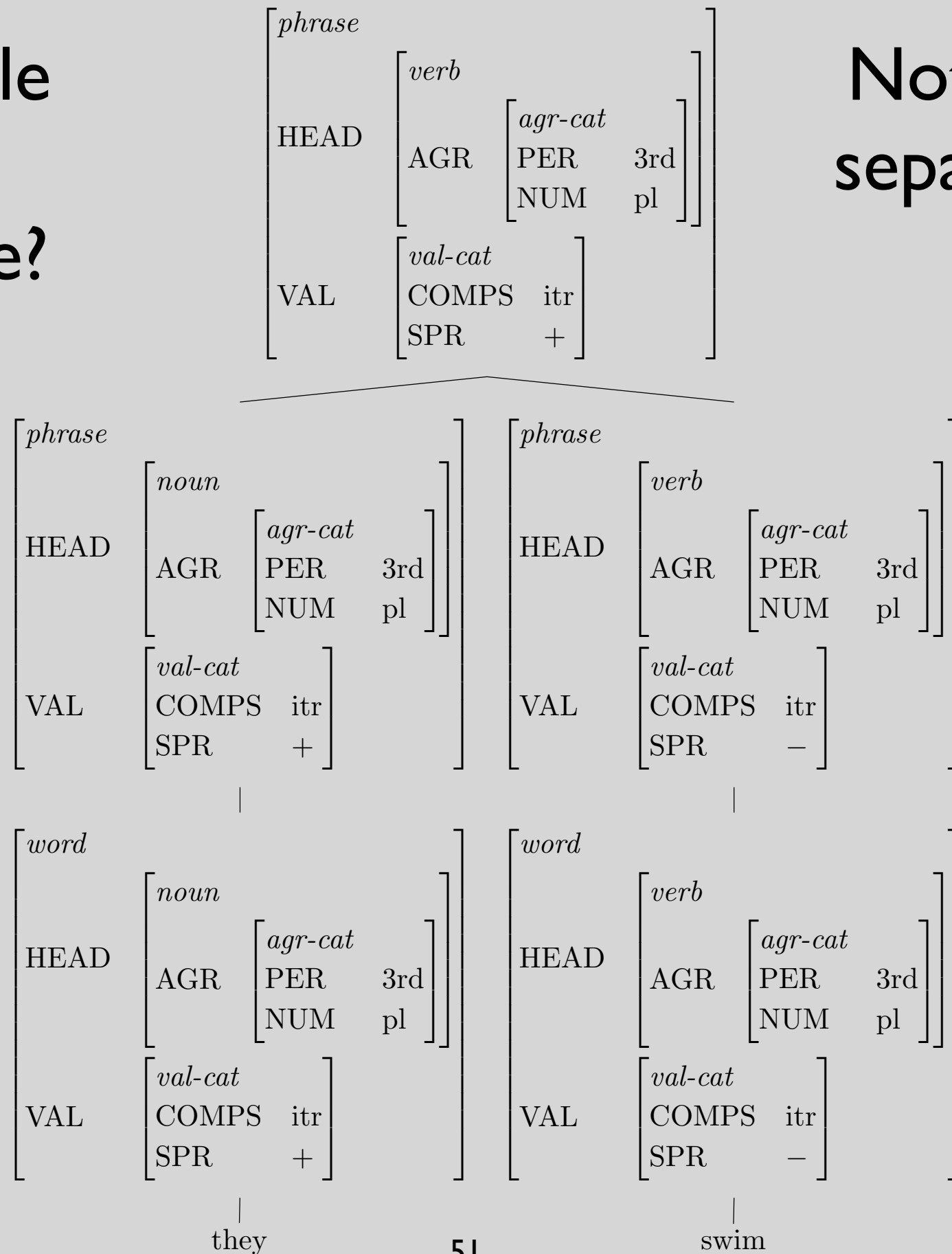
- It and each subtree are licensed by a grammar rule or lexical entry
- All general principles (like the HFP) are satisfied.
- NB: Trees are part of our model of the language, so all their features have values (even though we will often be lazy and leave out the values irrelevant to our current point).

Question:

Do phrases that are not headed have
HEAD features?

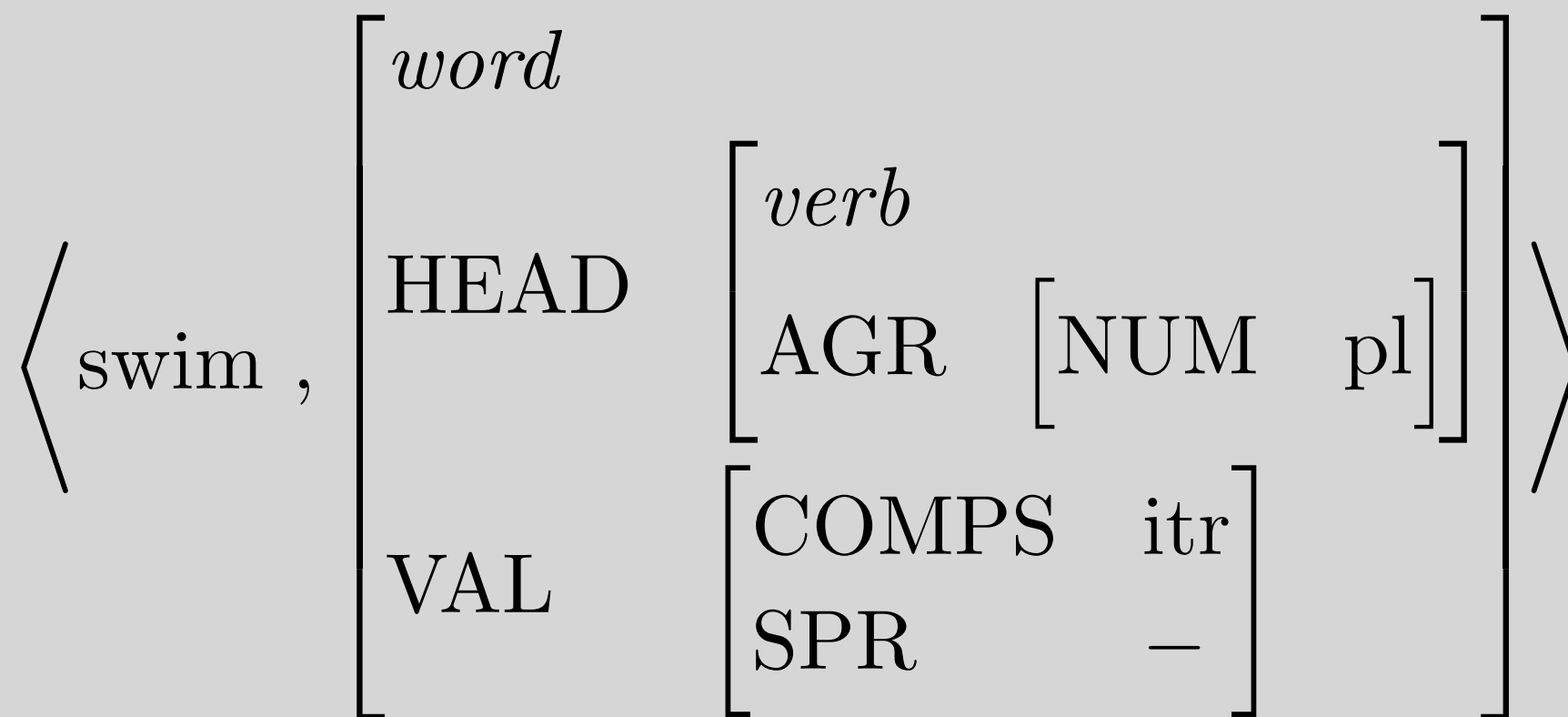
Which rule
licenses
each node?

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

RQs: Grammar changing

- I'm noticing that it seems like there are a lot of rules/ definitions/ processes that are created and then later replaced with something different/ better (e.g. atomic labels > feature structures > complex feature structures etc). I'm finding it a bit hard to keep track of what is changing/ what approaches have been discarded, specifically in terms of creating trees. As we learn more, and more of our process changes, is there an approach that you suggest to help keep track of this? And will any of the integral principles we use, like the HFP, be changing or can we consider these permanent?

RQs: COMPS/SPR

- Why can't COMPS just have values + and -?
- Why is the COMPS feature necessary for POS other than verbs? I thought it was only applicable to verbs.
- Why can't we leave N underspecified for COMPS if it is always itr?

RQs: COMPS/SPR

- Why are phrases "(whether NP,S,VP, or NOM) like intransitive verbs in that they cannot combine with complements"(p.65)? Is this another way to say the subtree they license is generated only by the mother node?

RQs: COMPS/SPR

- I'm just generally confused by the SPR tag, I know that it'll pop up again in the next chapter, but it would be nice to have a mini explanation before then! Does it do more than tell us whether a noun needs a det? It seems like it might turn out to be more complex.

RQs: word v. phrase

- "In the sense we intend here, however, 'word' refers to the category that the lexicon associates with a given form like *disappeared* and 'phrase' refers to the category that the grammar associates with a sequence of such forms."
- Why isn't this just the usual sense of "word" and "phrase"?

RQs: Type hierarchy

- What is the relationship between branches in the type hierarchy?

RQs: Underspecification

- What does this mean:
- “We no longer think of our phrase structure rules as specifying all the information that labels the nodes of trees (p83)”

RQs: Fully resolved trees

- Page 75 states that "each feature has a completely resolved value". What does it mean for a feature's value to be completely resolved?
- What is the purpose of labeling noun phrases as [COMPS itr]? Wouldn't they always be intransitive?

RQs: Multilingualism

- Is it possible to develop a multilingual model in this way? How do you strike the balance between these principles at the beginning? Would you start by developing as complete a grammar as possible for one language, or try to describe shared types first?