Ling 566
Oct 6, 2022

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*

Two Kinds of Language Models

- Speakers’ internalized knowledge (their grammar)
- Set of sentences in the language

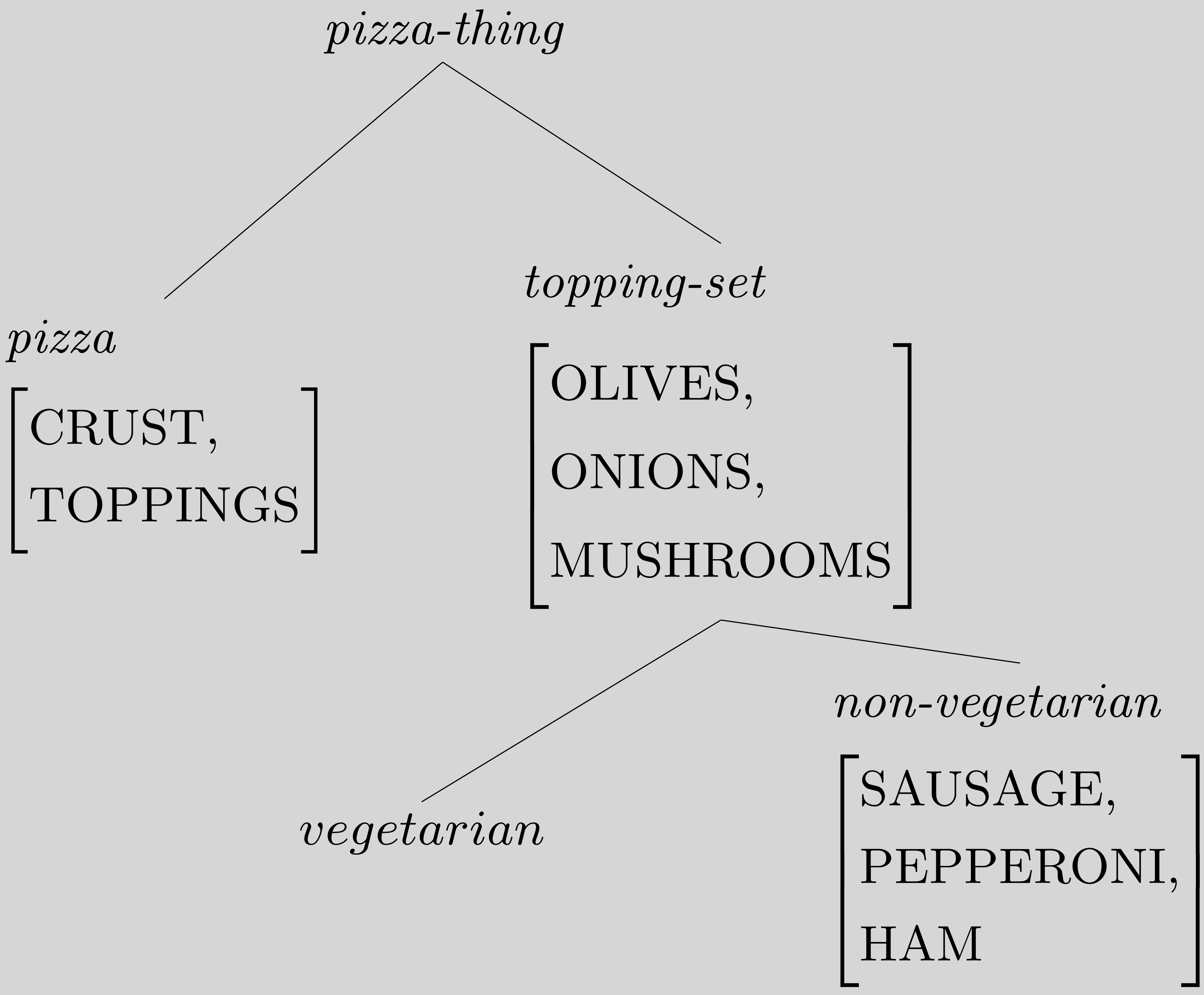
Things Involved in Modeling Language

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

Feature Structure Descriptions

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

A Pizza Type Hierarchy



TYPE	FEATURES/VALUES	IST
pizza-thing		
pizza	$\begin{bmatrix} \text{CRUST} & \{\text{thick, thin, stuffed}\} \\ \text{TOPPINGS} & \textit{topping-set} \end{bmatrix}$	*pizza-thing*
topping-set	$\begin{bmatrix} \text{OLIVES} & \{+, -\} \\ \text{ONIONS} & \{+, -\} \\ \text{MUSHROOMS} & \{+, -\} \end{bmatrix}$	*pizza-thing*
vegetarian		*topping-set*
non-vegetarian	$\begin{bmatrix} \text{SAUSAGE} & \{+, -\} \\ \text{PEPPERONI} & \{+, -\} \\ \text{HAM} & \{+, -\} \end{bmatrix}$	*topping-set*

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

$$\begin{bmatrix} \textit{pizza} & \\ \text{CRUST} & \text{thick} \\ \text{TOPPINGS} & \begin{bmatrix} \textit{vegetarian} & \\ \text{OLIVES} & + \\ \text{ONIONS} & + \end{bmatrix} \end{bmatrix}$$

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

Answer: 2

$$\begin{bmatrix} \textit{pizza} & \\ \text{CRUST} & \text{thick} \\ \text{TOPPINGS} & \begin{bmatrix} \textit{vegetarian} & \\ \text{OLIVES} & + \\ \text{ONIONS} & + \end{bmatrix} \end{bmatrix}$$

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

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

Pizza Descriptions and Pizza Models

$$\begin{bmatrix} \textit{pizza} & \\ \text{CRUST} & \text{thick} \\ \text{TOPPINGS} & \begin{bmatrix} \textit{vegetarian} & \\ \text{OLIVES} & + \\ \text{ONIONS} & + \end{bmatrix} \end{bmatrix}$$

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

Answer: A large, constantly-changing number.

Pizza Descriptions and Pizza Models

$$\begin{bmatrix} \textit{pizza} & \\ \text{CRUST} & \text{thick} \\ \text{TOPPINGS} & \begin{bmatrix} \textit{vegetarian} & \\ \text{OLIVES} & + \\ \text{ONIONS} & + \end{bmatrix} \end{bmatrix}$$

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

Combining Constraints

$$\begin{bmatrix} \textit{pizza} & \\ \text{CRUST} & \text{thick} \\ \text{TOPPINGS} & \begin{bmatrix} \text{OLIVES} & + \\ \text{HAM} & - \end{bmatrix} \end{bmatrix} \;\&\; \begin{bmatrix} \textit{pizza} & \\ \text{TOPPINGS} & \begin{bmatrix} \text{OLIVES} & + \\ \text{ONIONS} & + \end{bmatrix} \end{bmatrix}$$

Combining Constraints

$$\begin{bmatrix} \textit{pizza} & \\ \text{CRUST} & \text{thick} \\ \text{TOPPINGS} & \begin{bmatrix} \text{OLIVES} & + \\ \text{ONIONS} & + \\ \text{HAM} & - \end{bmatrix} \end{bmatrix}$$

Combining Constraints

$$\begin{bmatrix} \textit{pizza} & \\ \text{CRUST} & \text{thick} \\ \text{TOPPINGS} & \begin{bmatrix} \text{OLIVES} & + \\ \text{HAM} & - \end{bmatrix} \end{bmatrix} \;\&\; \begin{bmatrix} \textit{pizza} & \\ \text{CRUST} & \text{thin} \\ \text{TOPPINGS} & \begin{bmatrix} \text{OLIVES} & + \\ \text{ONIONS} & + \end{bmatrix} \end{bmatrix}$$

$$= \phi$$

Combining Constraints

$$\begin{bmatrix} \textit{pizza} & \\ \text{CRUST} & \text{thick} \\ \text{TOPPINGS} & \begin{bmatrix} \text{OLIVES} & + \\ \text{HAM} & + \end{bmatrix} \end{bmatrix} \;\&\; \begin{bmatrix} \textit{pizza} & \\ \text{CRUST} & \text{thick} \\ \text{TOPPINGS} & \textit{vegetarian} \end{bmatrix}$$

$$= \phi$$

Combining Constraints

$$\begin{bmatrix} \mathit{pizza} & \\ \text{CRUST} & \text{thick} \\ \text{TOPPINGS} & \begin{bmatrix} \text{OLIVES} & + \\ \text{HAM} & - \end{bmatrix} \end{bmatrix} \;\&\; \begin{bmatrix} \mathit{pizza} & \\ \text{CRUST} & \text{thick} \\ \text{TOPPINGS} & \mathit{vegetarian} \end{bmatrix}$$

$$= \phi$$

A New Theory of Pizzas

$$\textit{pizza} : \begin{bmatrix} \text{CRUST} & \left\{ \text{thick} \text{ , } \text{thin} \text{ , } \text{stuffed} \right\} \\ \text{ONE-HALF} & \textit{topping-set} \\ \text{OTHER-HALF} & \textit{topping-set} \end{bmatrix}$$

Combining Constraints

$$\begin{bmatrix} \textit{pizza} & \\ \text{ONE-HALF} & \begin{bmatrix} \text{ONIONS} & + \\ \text{OLIVES} & - \end{bmatrix} \end{bmatrix} \;\&\; \begin{bmatrix} \textit{pizza} & \\ \text{OTHER-HALF} & \begin{bmatrix} \text{ONIONS} & - \\ \text{OLIVES} & + \end{bmatrix} \end{bmatrix}$$

$$=$$

$$\begin{bmatrix} \textit{pizza} & \\ \text{ONE-HALF} & \begin{bmatrix} \text{ONIONS} & + \\ \text{OLIVES} & - \end{bmatrix} \\ \text{OTHER-HALF} & \begin{bmatrix} \text{ONIONS} & - \\ \text{OLIVES} & + \end{bmatrix} \end{bmatrix}$$

Identity Constraints (tags)

$$\begin{bmatrix} \textit{pizza} & \\ \text{CRUST} & \text{thin} \\ \text{ONE-HALF} & \begin{bmatrix} \text{OLIVES} & \boxed{1} \\ \text{ONIONS} & \boxed{2} \end{bmatrix} \\ \text{OTHER-HALF} & \begin{bmatrix} \text{OLIVES} & \boxed{1} \\ \text{ONIONS} & \boxed{2} \end{bmatrix} \end{bmatrix}$$

Combining Constraints

$$\begin{bmatrix} \textit{pizza} & \\ \text{ONE-HALF} & \boxed{1}\begin{bmatrix}\text{ONIONS} & + \\ \text{OLIVES} & -\end{bmatrix} \\ \text{OTHER-HALF} & \boxed{1} \end{bmatrix} \; \& \; \begin{bmatrix} \textit{pizza} & \\ \text{OTHER-HALF} & \begin{bmatrix}\text{MUSHROOMS} & - \\ \text{OLIVES} & -\end{bmatrix} \end{bmatrix}$$

$$=$$

$$\begin{bmatrix} \textit{pizza} & \\ \text{ONE-HALF} & \boxed{1}\begin{bmatrix}\text{ONIONS} & + \\ \text{OLIVES} & - \\ \text{MUSHROOMS} & -\end{bmatrix} \\ \text{OTHER-HALF} & \boxed{1} \end{bmatrix}$$

Note

$$\begin{bmatrix} \textit{pizza} & \\ \text{ONE-HALF} & \boxed{1}\begin{bmatrix}\text{ONIONS} & + \\ \text{OLIVES} & - \\ \text{MUSHROOMS} & -\end{bmatrix} \\ \text{OTHER-HALF} & \boxed{1} \end{bmatrix}$$

$$=$$

$$\begin{bmatrix} \textit{pizza} & \\ \text{ONE-HALF} & \boxed{1} \\ \text{OTHER-HALF} & \boxed{1}\begin{bmatrix}\text{ONIONS} & + \\ \text{OLIVES} & - \\ \text{MUSHROOMS} & -\end{bmatrix} \end{bmatrix}$$

Combining Constraints

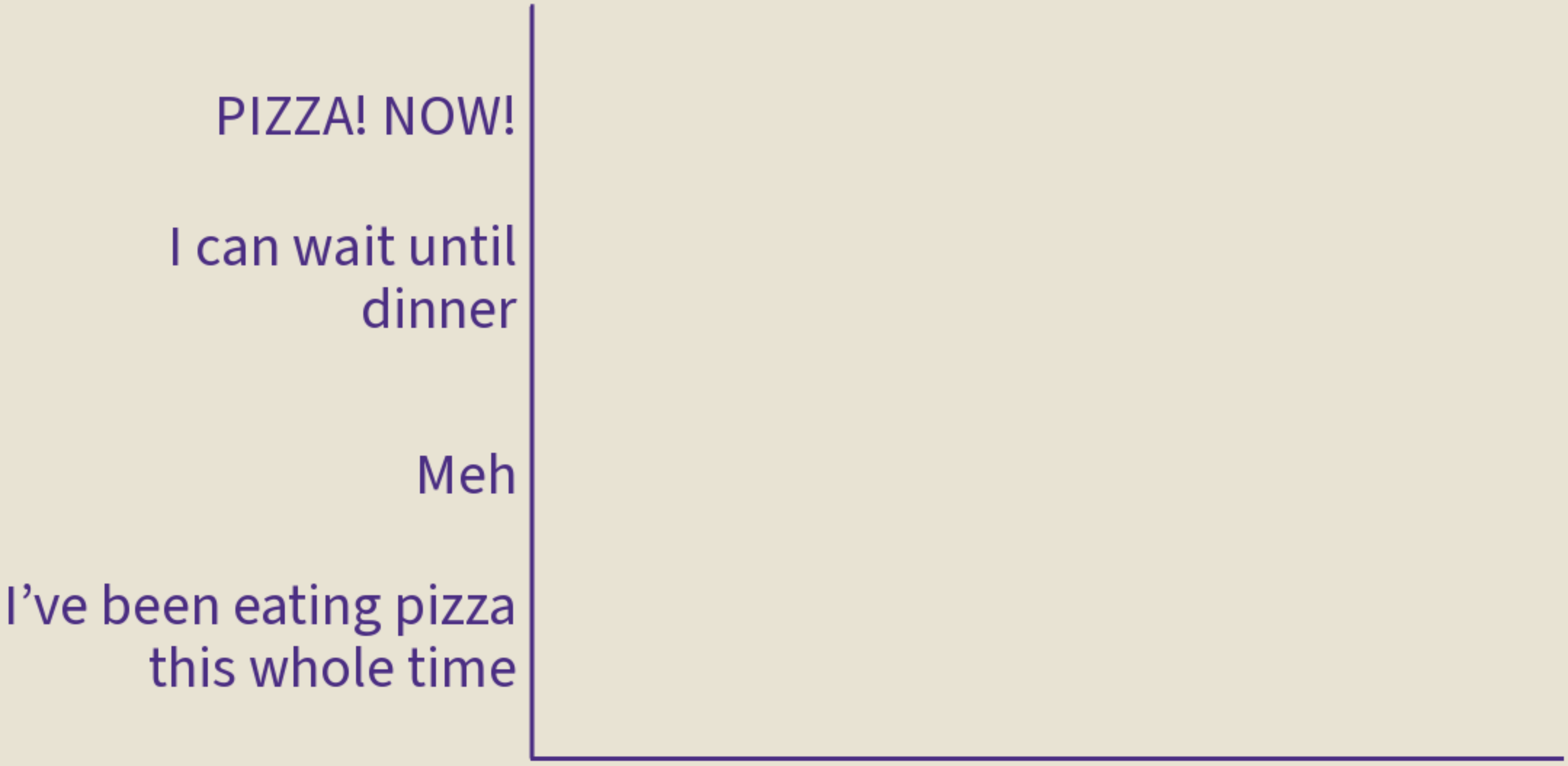
$$\begin{bmatrix} \textit{pizza} & \\ \text{ONE-HALF} & \boxed{1}\begin{bmatrix} \text{ONIONS} & + \\ \text{OLIVES} & + \end{bmatrix} \\ \text{OTHER-HALF} & \boxed{1}\ \textit{vegetarian} \end{bmatrix} \ \& \ \begin{bmatrix} \textit{pizza} & \\ \text{ONE-HALF} & \begin{bmatrix} \text{SAUSAGE} & + \\ \text{HAM} & - \end{bmatrix} \end{bmatrix}$$

$$= \phi$$

When poll is active, respond at **pollev.com/emb**

Text **EMB** to **22333** once to join

W How badly do you want pizza now?



Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Why combine constraints?

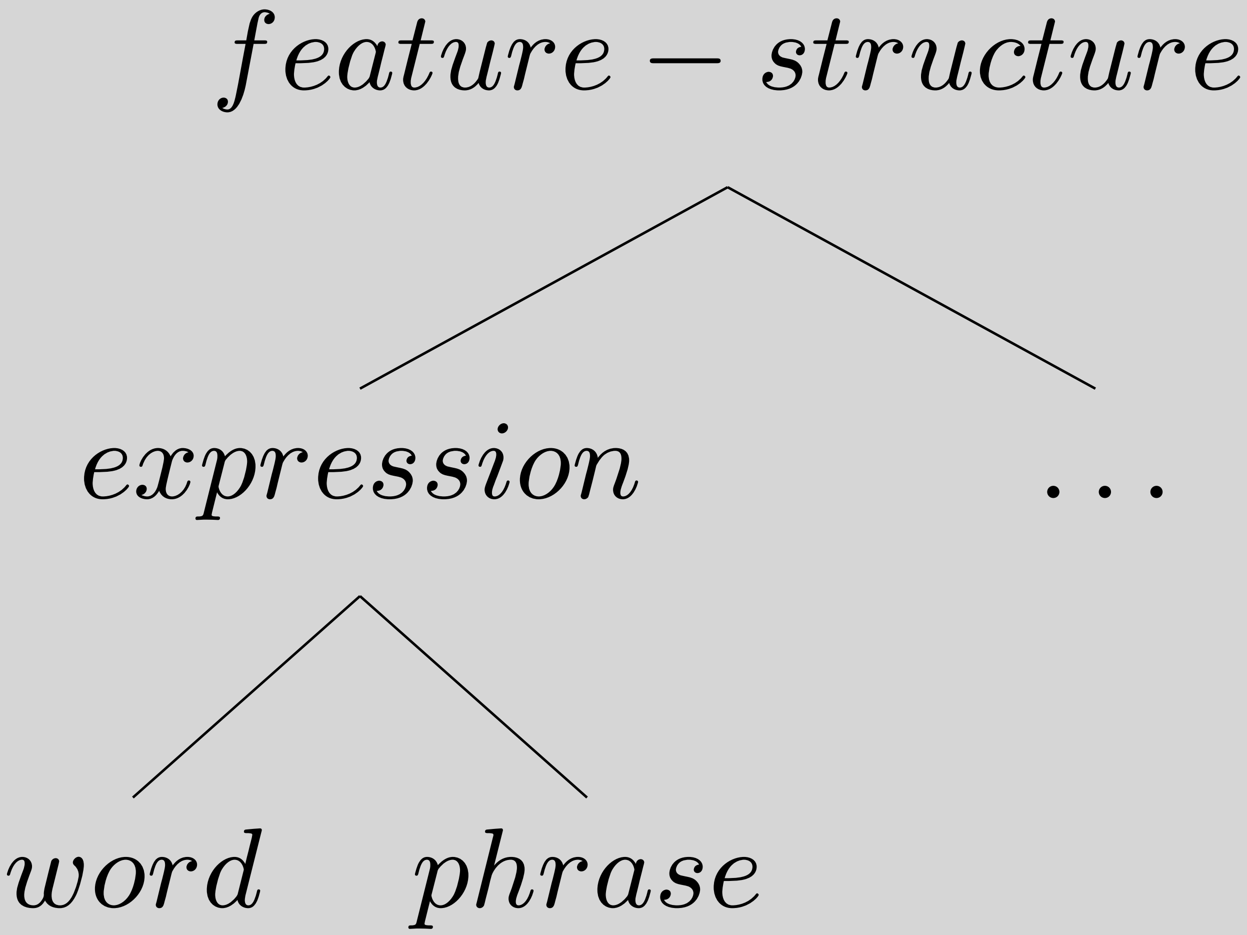
- The pizza example illustrates how unification can be used to combine information from different sources.
- In our grammar, information will come from lexical entries, grammar rules, and general principles.

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



feature – structure

expression

. . .

word

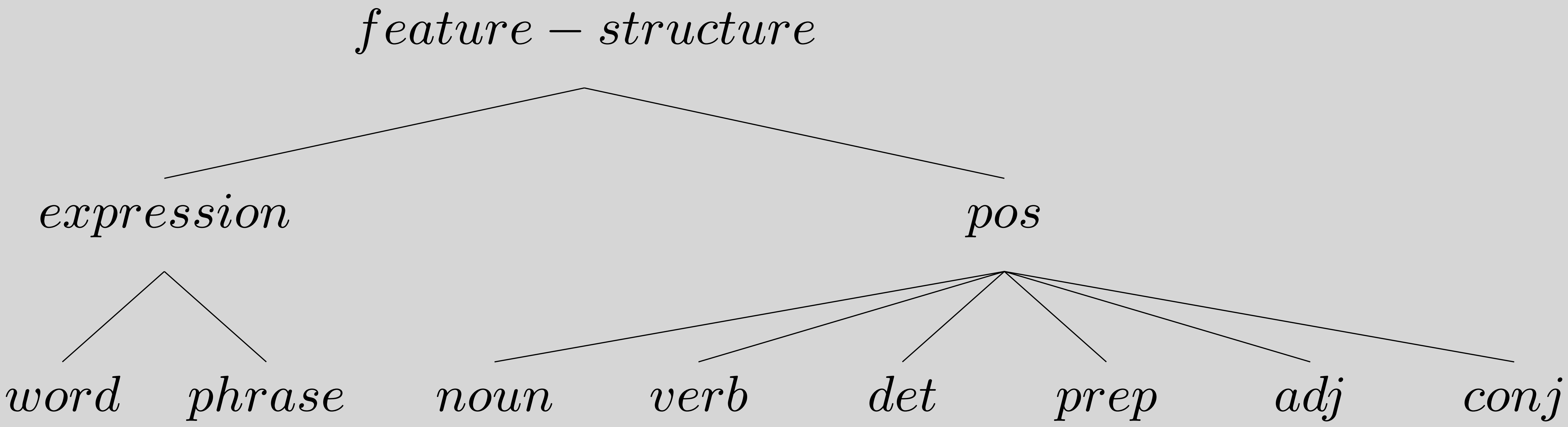
phrase

A Feature for Part of Speech

$$\text{NP} = \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{noun} \end{bmatrix}$$

$$\left\langle \text{bird}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{noun} \end{bmatrix} \right\rangle$$

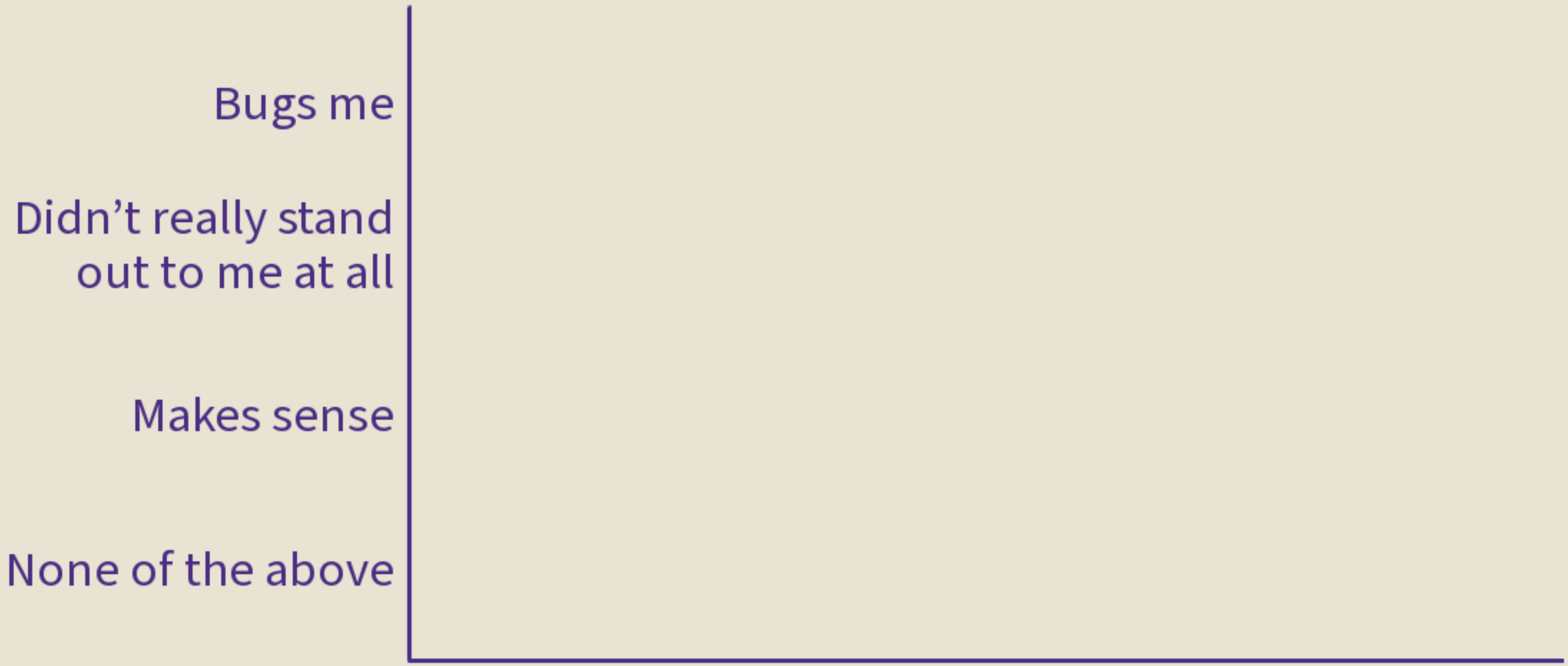
Type Hierarchy for Parts of Speech I



When poll is active, respond at **pollev.com/emb**

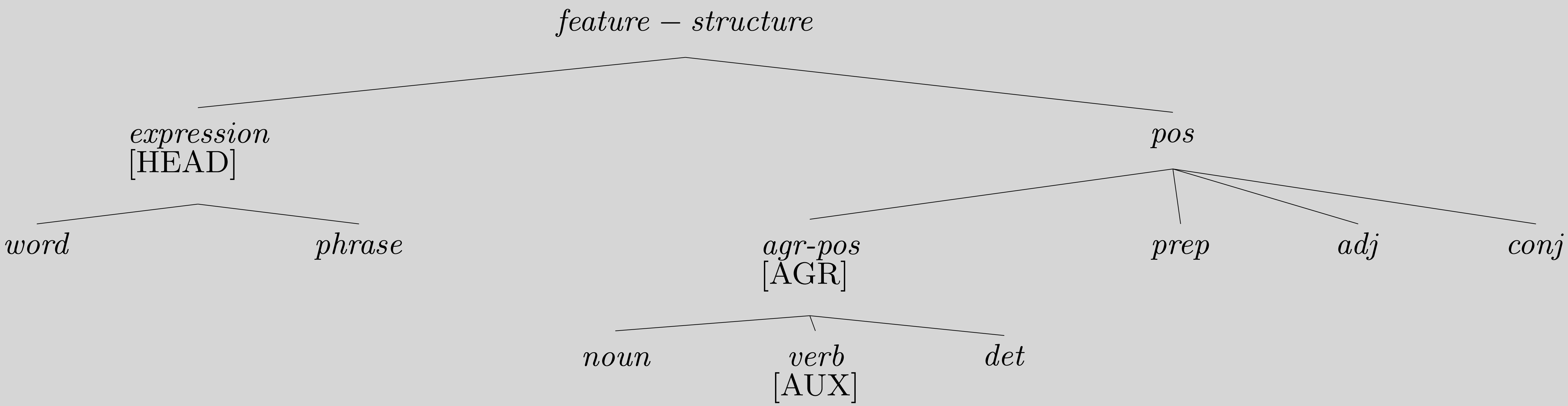
Text **EMB** to **22333** once to join

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



Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Type Hierarchy for Parts of Speech II



A Feature for Valence

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

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

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

Underspecification

$$\mathrm{V} \;=\; \begin{bmatrix} \textit{word} & \\ \mathrm{HEAD} & \textit{verb} \end{bmatrix}$$

$$\mathrm{VP} \;=\; \begin{bmatrix} \textit{phrase} & \\ \mathrm{HEAD} & \textit{verb} \end{bmatrix}$$

$$\begin{bmatrix} \mathrm{HEAD} & \textit{verb} \end{bmatrix}$$

Another Valence Feature

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

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

SPR and Verbs

$$\mathrm{S} = \begin{bmatrix} \textit{phrase} & \\ \mathrm{HEAD} & \textit{verb} \\ \mathrm{VAL} & \begin{bmatrix} \mathrm{COMPS} & \mathrm{itr} \\ \mathrm{SPR} & + \end{bmatrix} \end{bmatrix}$$

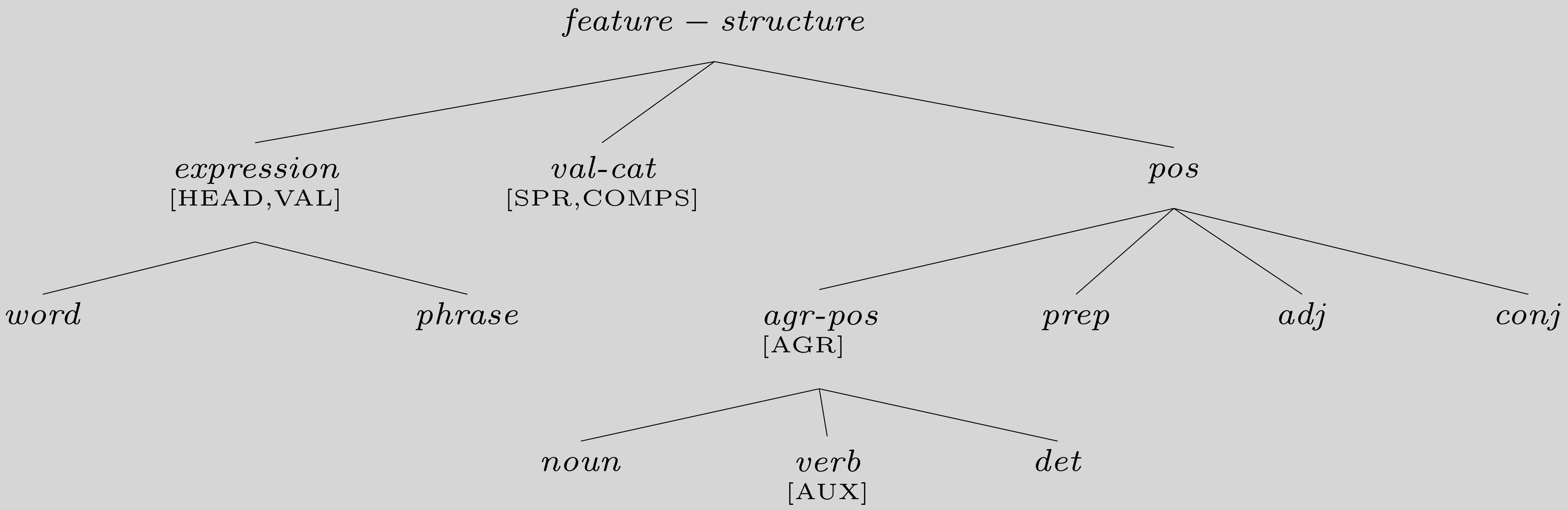
$$\mathrm{VP} = \begin{bmatrix} \textit{phrase} & \\ \mathrm{HEAD} & \textit{verb} \\ \mathrm{VAL} & \begin{bmatrix} \mathrm{COMPS} & \mathrm{itr} \\ \mathrm{SPR} & - \end{bmatrix} \end{bmatrix}$$

S and NP

$$\begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix}$$

- We created a monster
- our creation of a monster

Type Hierarchy So Far



Reading Questions

- I'm a bit confused by the Type Hierarchy outlined in (69). Specifically, what are the branching relationships? I'm thrown by the fact that HEAD and VAL appear at the same level, but in a separate branch, from the values they can take (val-cat and pos).

Reformulating the Grammar Rules I
Which Ch 2 rules do these correspond to?

Head-Complement Rule 1:

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

Head Complement Rule 2:

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

Head Complement Rule 3:

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

Reformulating the Grammar Rules II

Head-Specifier Rule 1:

$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \rightarrow \overset{\text{NP}}{\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \end{bmatrix}} \quad \mathbf{H}\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

Head-Specifier Rule 2:

$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \rightarrow \text{D} \quad \mathbf{H}\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

Reformulating the Grammar Rules III

Non-Branching NP Rule

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

Head-Modifier Rule

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

Coordination Rule

$$\boxed{1} \rightarrow \boxed{1}^{+} \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{conj} \end{bmatrix} \boxed{1}$$

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 …

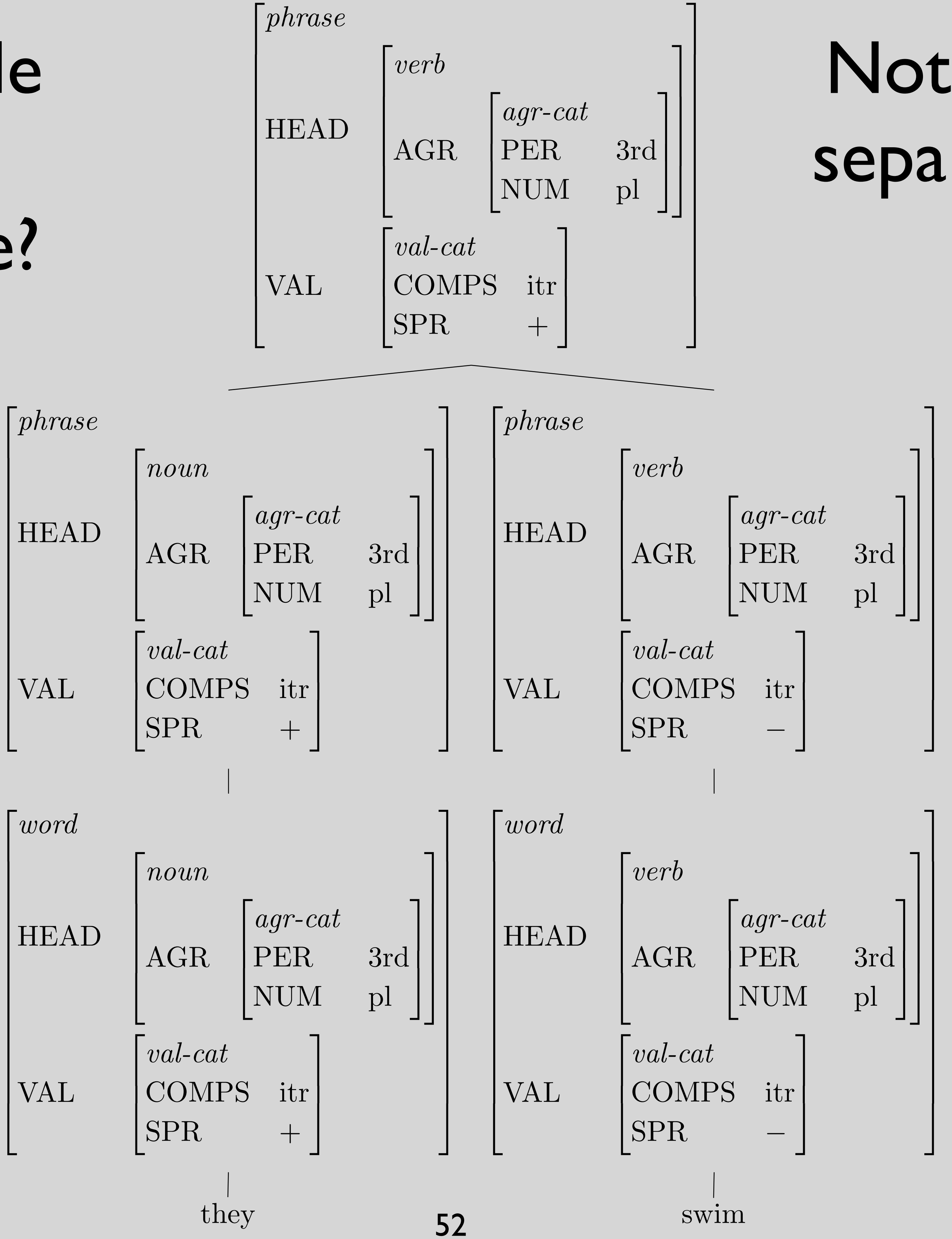
- It and each subtree are licensed by a grammar rule or lexical entry
- All general principles (like the HFP) are satisfied.
- NB: Trees are part of our model of the language, so all their features have values (even though we will often be lazy and leave out the values irrelevant to our current point).

Question:

Do phrases that are not headed have HEAD features?

Which rule licenses each node?

Note the three separate uses of DAGs



$$\left[\begin{array}{ll}\textit{phrase} & \\ \text{HEAD} & \left[\begin{array}{ll}\textit{verb} & \\ \text{AGR} & \left[\begin{array}{ll}\textit{agr-cat} & \\ \text{PER} & \text{3rd}\\ \text{NUM} & \text{pl}\end{array}\right]\end{array}\right]\\ \text{VAL} & \left[\begin{array}{ll}\textit{val-cat} & \\ \text{COMPS} & \text{itr}\\ \text{SPR} & +\end{array}\right]\end{array}\right]$$

$$\left[\begin{array}{ll}\textit{phrase} & \\ \text{HEAD} & \left[\begin{array}{ll}\textit{noun} & \\ \text{AGR} & \left[\begin{array}{ll}\textit{agr-cat} & \\ \text{PER} & \text{3rd}\\ \text{NUM} & \text{pl}\end{array}\right]\end{array}\right]\\ \text{VAL} & \left[\begin{array}{ll}\textit{val-cat} & \\ \text{COMPS} & \text{itr}\\ \text{SPR} & +\end{array}\right]\end{array}\right] \quad \left[\begin{array}{ll}\textit{phrase} & \\ \text{HEAD} & \left[\begin{array}{ll}\textit{verb} & \\ \text{AGR} & \left[\begin{array}{ll}\textit{agr-cat} & \\ \text{PER} & \text{3rd}\\ \text{NUM} & \text{pl}\end{array}\right]\end{array}\right]\\ \text{VAL} & \left[\begin{array}{ll}\textit{val-cat} & \\ \text{COMPS} & \text{itr}\\ \text{SPR} & -\end{array}\right]\end{array}\right]$$

$$\left[\begin{array}{ll}\textit{word} & \\ \text{HEAD} & \left[\begin{array}{ll}\textit{noun} & \\ \text{AGR} & \left[\begin{array}{ll}\textit{agr-cat} & \\ \text{PER} & \text{3rd}\\ \text{NUM} & \text{pl}\end{array}\right]\end{array}\right]\\ \text{VAL} & \left[\begin{array}{ll}\textit{val-cat} & \\ \text{COMPS} & \text{itr}\\ \text{SPR} & +\end{array}\right]\end{array}\right] \quad \left[\begin{array}{ll}\textit{word} & \\ \text{HEAD} & \left[\begin{array}{ll}\textit{verb} & \\ \text{AGR} & \left[\begin{array}{ll}\textit{agr-cat} & \\ \text{PER} & \text{3rd}\\ \text{NUM} & \text{pl}\end{array}\right]\end{array}\right]\\ \text{VAL} & \left[\begin{array}{ll}\textit{val-cat} & \\ \text{COMPS} & \text{itr}\\ \text{SPR} & -\end{array}\right]\end{array}\right]$$

they swim

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?

$$\left\langle \text{swim} , \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \right\rangle$$

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

- Is there anything syntactically that's unwieldy or inconvenient to model with a feature structure? And are there commonly other types of data (the text briefly mentioned a phonological description) that might be captured within one, or is that only for specific use cases?

- How do we know a particular value for any feature is atomic and can't be further subdivided? This is in regard to section 3.3.6 where the claim PER and NUM are both atomic is presented without proof.

Reading Questions

- The book talks about models being complete (3.4.1), but lexical entries, grammar rules, and principles aren't usually fully resolved. How do you know what level of resolved-ness you need for something?
- Is it correct that each word or phrase corresponds to one feature structure, but each feature structure can correspond to many words or phrases?

Reading Questions

- When should we use tagging? For example, in (74) Head Specifier Rule 1 we are tagging for the same agreement on the right side, but make no mention of agreement on the left side. This is opposed to (68) when agreement is specified for the NP and not the N. Another question I have is if we mention the agreement on the left side of the rule, and never mention it on the right, are we just assuming it's all the same agreement? What is the use of tagging then?

Reading Questions

- N and DET seem to have the COMPS feature as well. Based on what do we decide the COMPS value of them? (On current scope, the values of COMPS (STR/ITR/DTR) seem to be defined only for verbs.)

- Why does the feature structure for D have COMPS itr? I would expect that the feature structure for D would have COMPS str since words such as the, those, etc. expect a complement, but (45) shows the feature structure for D has COMPS itr. If my intuition about the meaning of COMPS is incorrect then what does it mean in the context of nouns?

Reading Questions

- Why maintain the “phrase”, “word”, and “expression” types? Wouldn’t the specific linguistic category (i.e. S, VP, NP, etc) be more straightforward? I’m not fully understanding the benefit of abstracting to “phrase”, “word”, and “expression”.