Ling 566
Oct 7, 2021

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

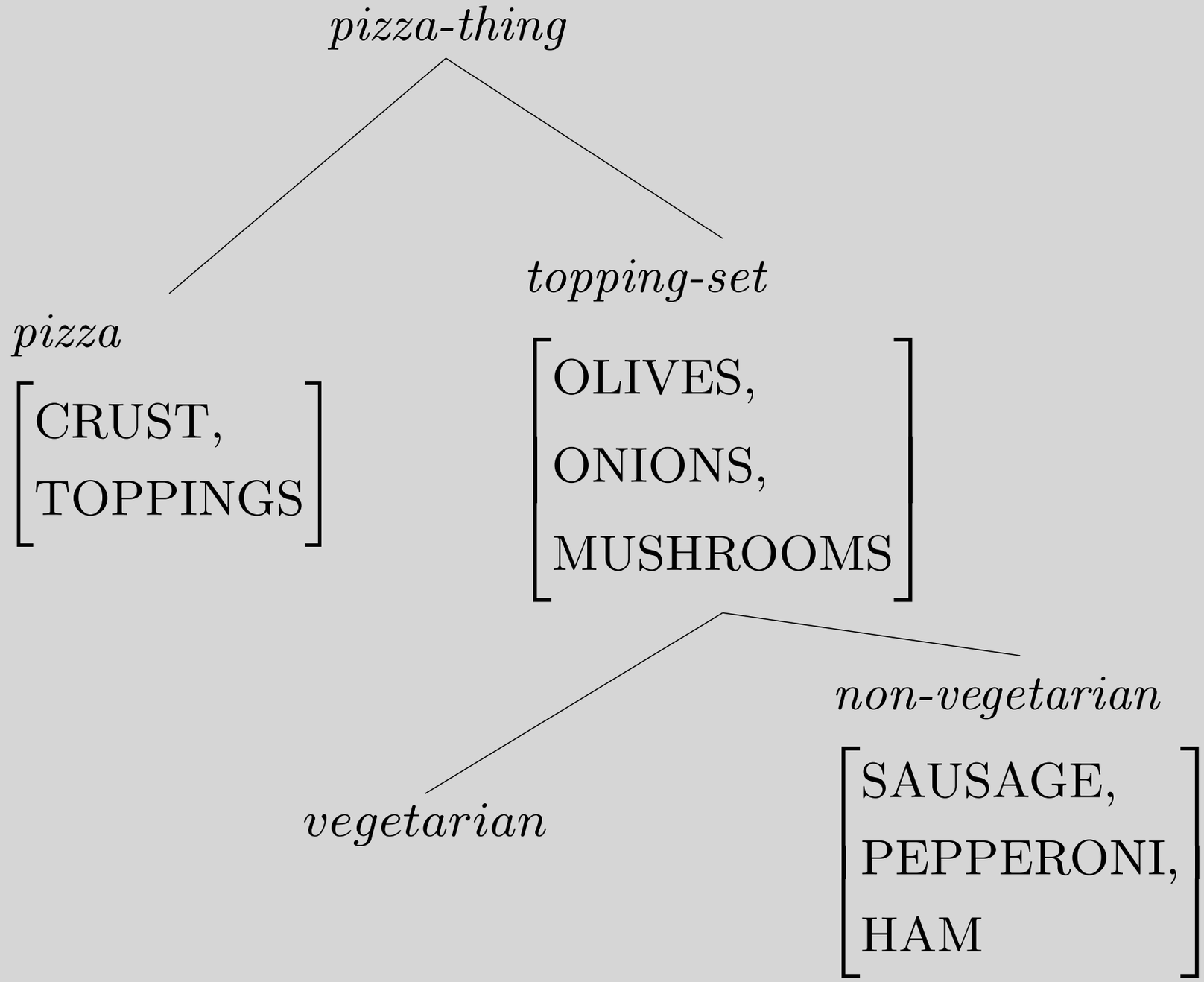
Things Involved in Modeling Language

- 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

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

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

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

$$= \phi$$

Combining Constraints

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

$$= \phi$$

Combining Constraints

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

$$= \phi$$

A New Theory of Pizzas

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

Combining Constraints

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

$$=$$

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

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

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

$$=$$

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

Note

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

$$=$$

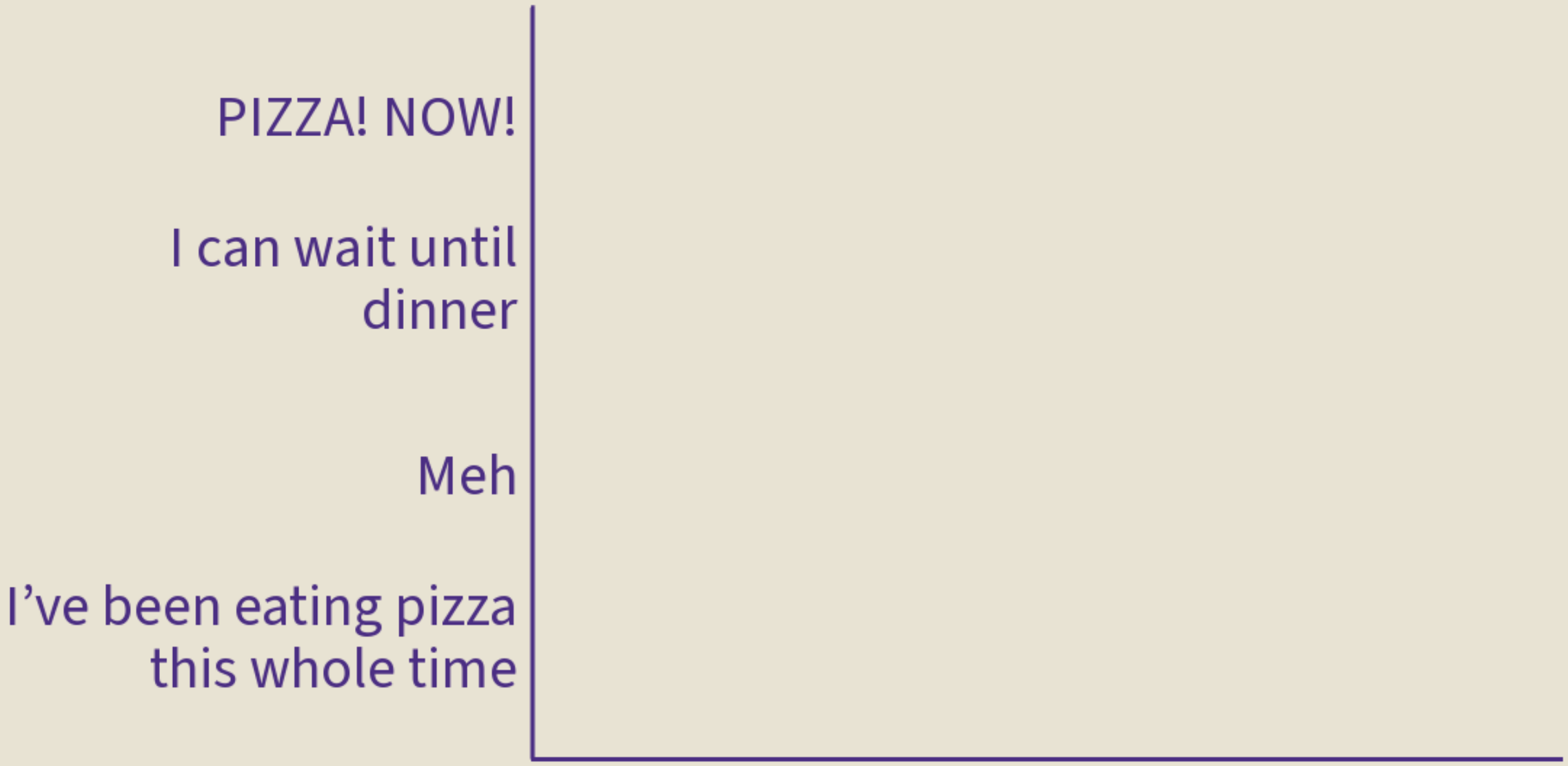
$$\begin{bmatrix} \textit{pizza} & \\ \text{ONE-HALF} & \boxed{1} \\ \text{OTHER-HALF} & \boxed{1}\begin{bmatrix} \text{ONIONS} & + \\ \text{OLIVES} & - \\ \text{MUSHROOMS} & - \end{bmatrix} \end{bmatrix}$$

Combining Constraints

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

$$= \phi$$

How badly do you want pizza now?



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?

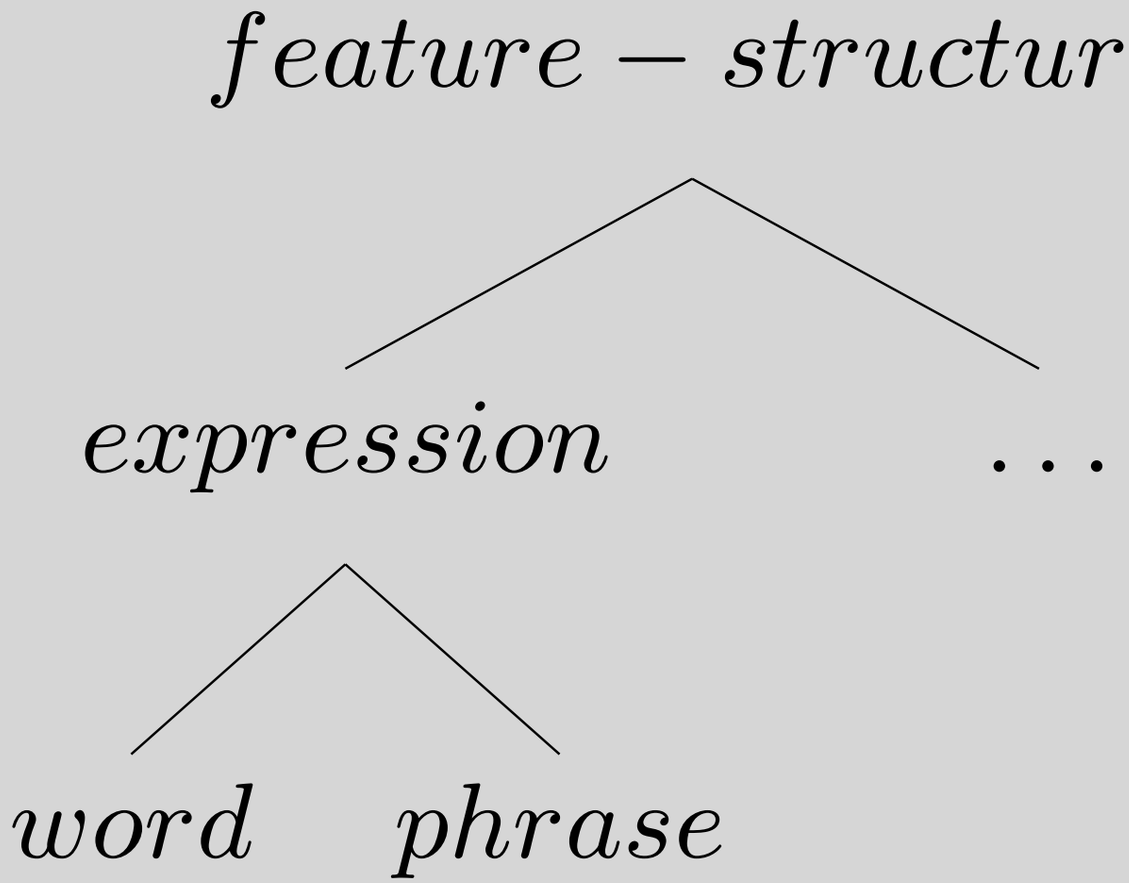
NP & VP: are both phrases

N & V: are both words

NP & N: are both ‘nouny’

VP & V: are both ‘verby’

The Beginnings of Our Type Hierarchy

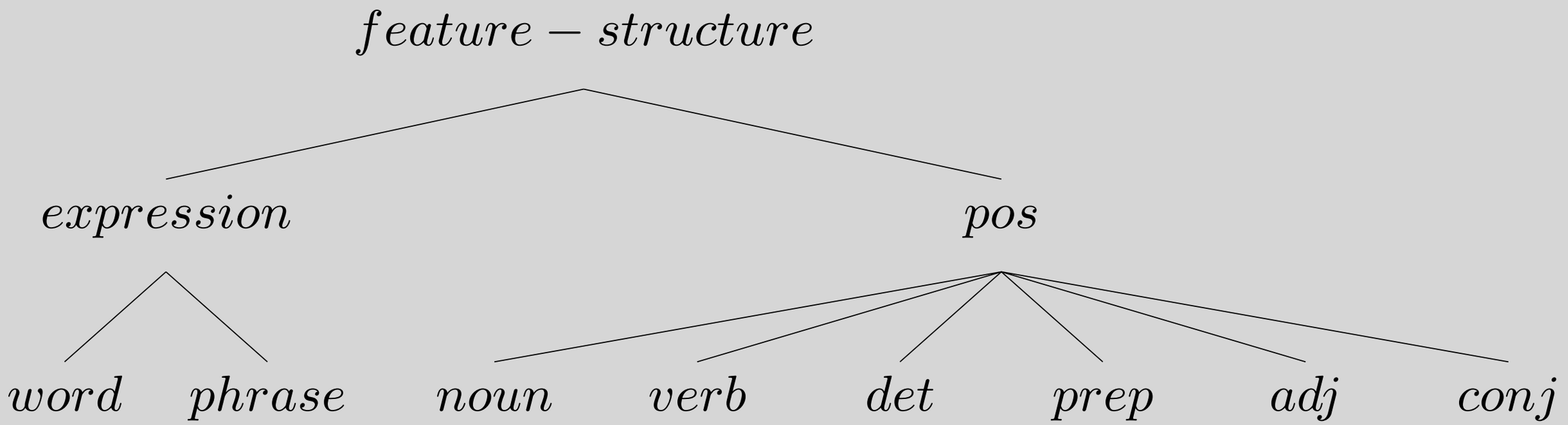


A Feature for Part of Speech

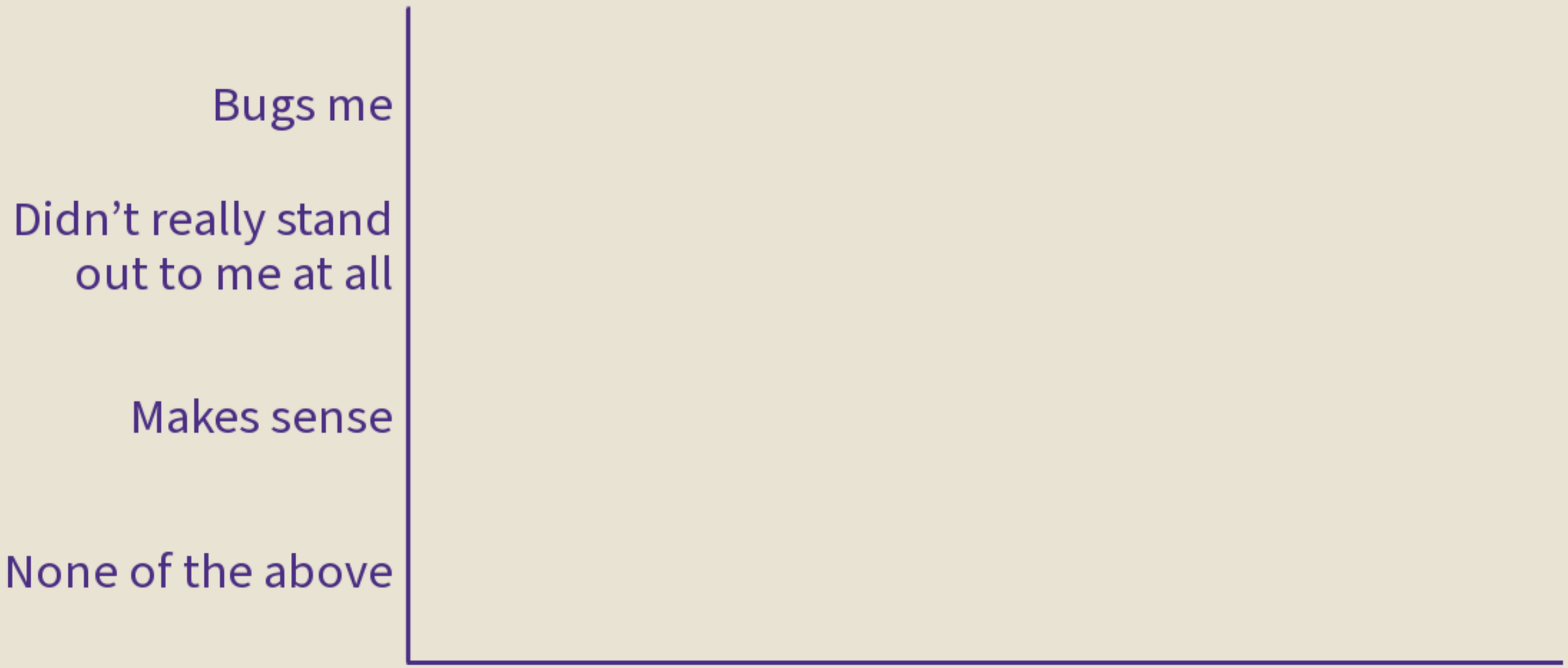
$$\text{NP} = \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{noun} \end{bmatrix}$$

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

Type Hierarchy for Parts of Speech I



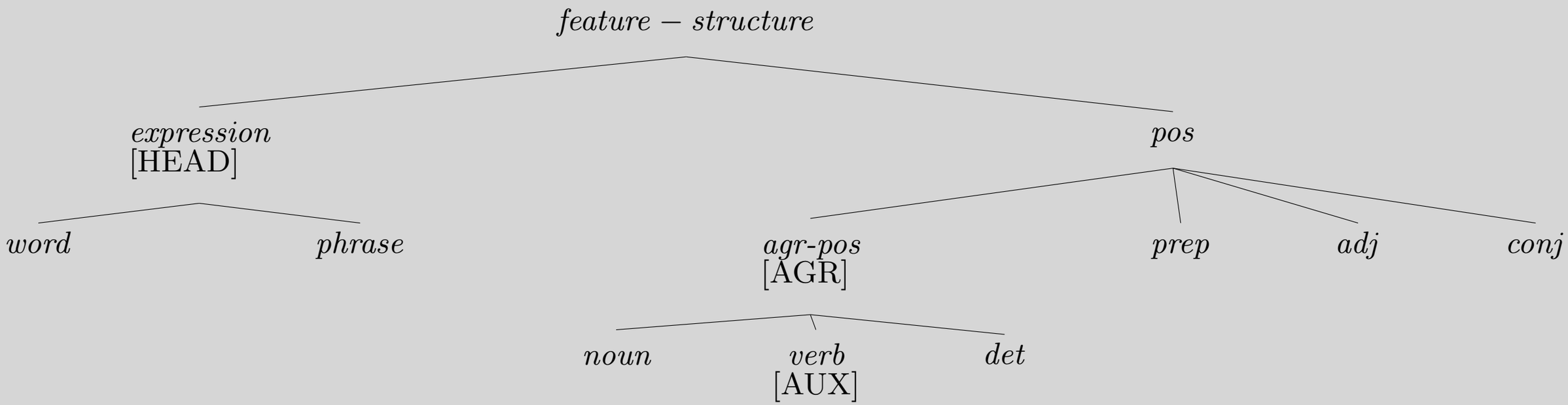
W Have 'expression' and 'pos' at the same level of that hierarchy



Reading Questions

- I am confused with why pos is at the same level with expressions and not under word?

Type Hierarchy for Parts of Speech II



A Feature for Valence

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

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

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

Underspecification

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

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

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

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

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

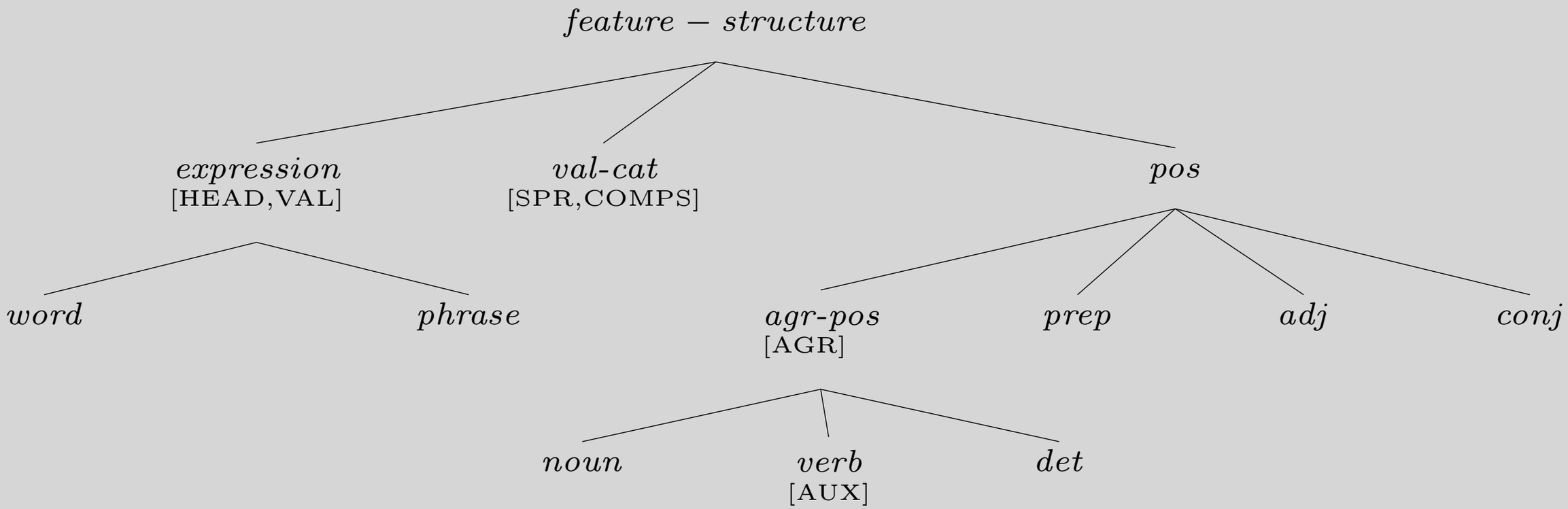
$$\mathrm{VP} = \begin{bmatrix} \textit{phrase} & \\ \mathrm{HEAD} & \textit{verb} \\ \mathrm{VAL} & \begin{bmatrix} \mathrm{COMPS} & \mathrm{itr} \\ \mathrm{SPR} & - \end{bmatrix} \end{bmatrix}$$

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



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

Head-Complement Rule 1:

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

Head Complement Rule 2:

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

Head Complement Rule 3:

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

Reformulating the Grammar Rules II

Head-Specifier Rule 1:

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

Head-Specifier Rule 2:

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

Reformulating the Grammar Rules III

Non-Branching NP Rule

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

Head-Modifier Rule

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

Coordination Rule

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

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

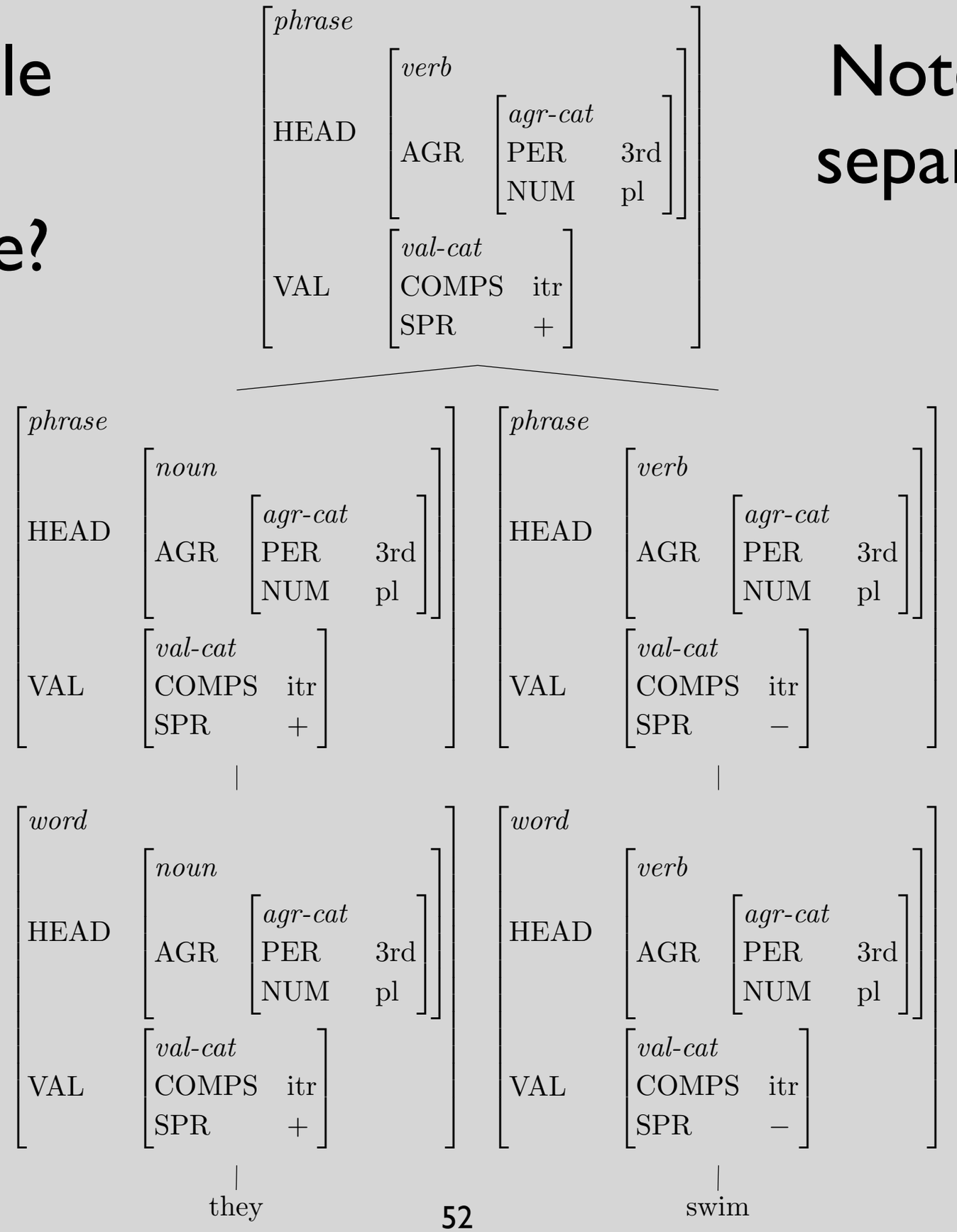
A Tree is Well-Formed if …

- 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

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?

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

Reading Question

- Can you go over (56) on page 75 and why underspecifying NUM licenses two distinct WORD Structures and how do we decide on leaving a feature underspecified or not?

- (56) is a lexical entry for *you*, similarly underspecified for number.

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

- What is a head?
- of a rule
- of a phrase
- as a feature

Reading Questions

- I feel like I'm still unsure how we know that VP is the head of S. How do we know which daughter is the head daughter?
- Is there flexibility in the daughter node we choose as the "head daughter" for a particular headed rule? Additionally, would there be any utility in allowing a headed phrase to have multiple headed daughters?
- Which daughter is the head of *the dog*?

Reading Questions

- Why does the VAL feature structure mentioned in section 3.3.4 need a feature type? When should a new feature type be created and when not?
- What is underspecification and why do we need it?
- In the example in section 3.4.2 (pages 78-81), I am confused as to what the small changes in the feature structure signify. Namely, why do the lexical entries not have the agr-cat and val-cat labels but the feature structure versions of them do? In other words, I am still unclear about the function and meaning of the categories "agr-cat" and "val-cat". How are they different from AGR and VAL?

Reading Questions

- p.73: "By formulating HFP in terms of HEAD value identity, we allow information specified by the rule, information present on the daughter or the mother, or information required by some other constraint all to be amalgamated..." I'm not sure I understand what this means or how the HFP allows for all this.

Reading Questions

- p.62: "... that it is not the words themselves that specify values for these features [like AGR and AUX] ..." but rather that the "Individual words (and phrases) get associated with this information because they have a feature HEAD whose value is always a feature structure that belongs to some subtype of pos.” Isn't the word('s meaning) what provides the feature structure that is then used to specify the values of its feature structure? In this case, what is the reason for establishing a word and its feature structure as adjacent and associated rather than directional (where the word determines the feature structure / the feature structure is derived from the word)? Does this distinction have more to do with semantics than syntax?

Reading Questions

- In Section 3.3.5 (p. 69), the authors claim that the head daughter's COMPS value can be omitted given that its type is phrase because all categories of type phrase have [COMPS ite]. Does this imply that we don't need to specify COMPS for any category of phrase type? For example, can we also make COMPS underspecified for the VP/NOM structure in the diagram (47)? Besides, I am not very clear why COMPS is applied to all categories rather than made specific for V and VP which makes more sense to me.