Ling 566
Oct 3, 2019

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

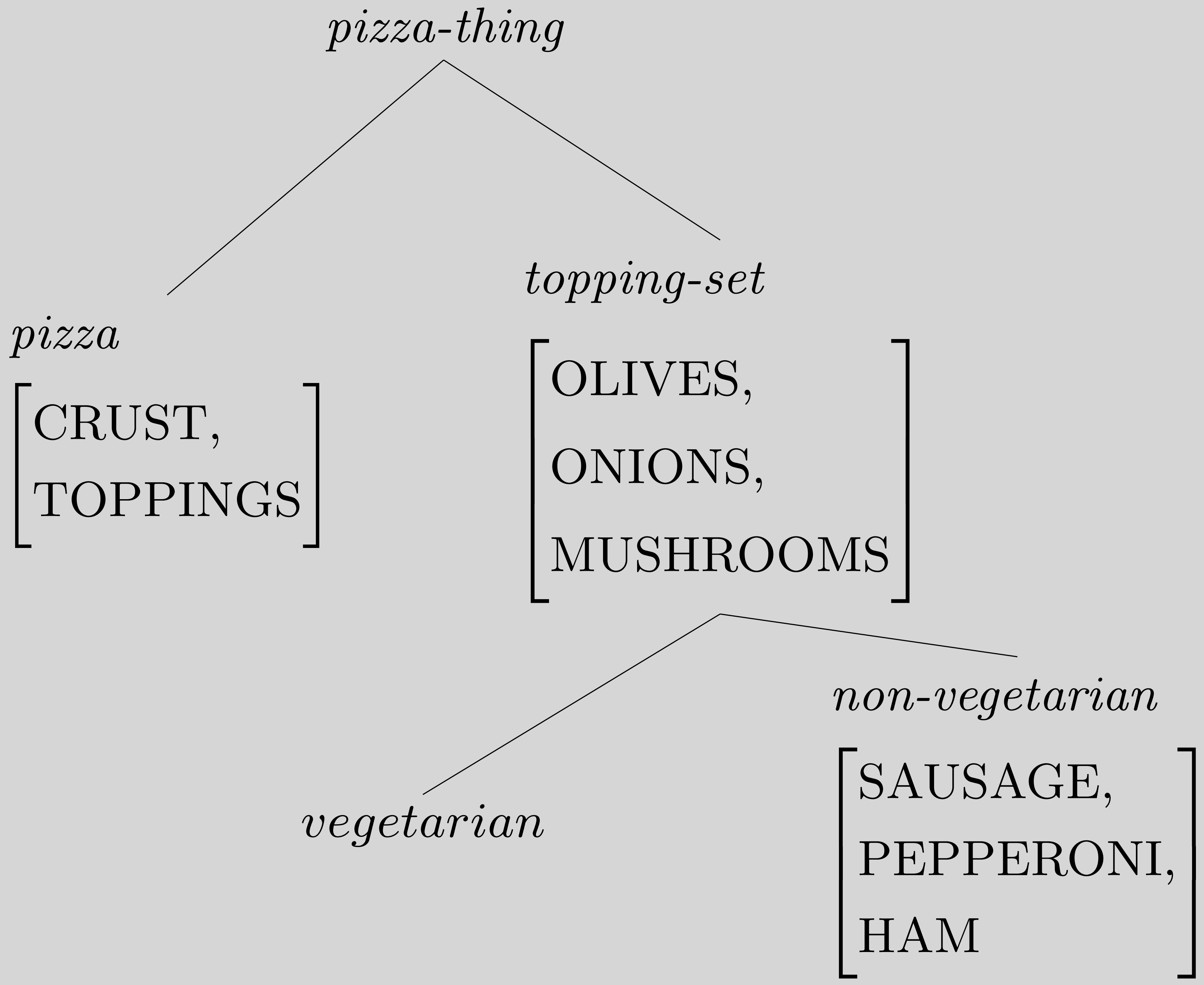
Things Involved in Modeling Language

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

Feature Structure Descriptions

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

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

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

$$= \phi$$

A New Theory of Pizzas

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

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

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

$$= \phi$$

Why combine constraints?

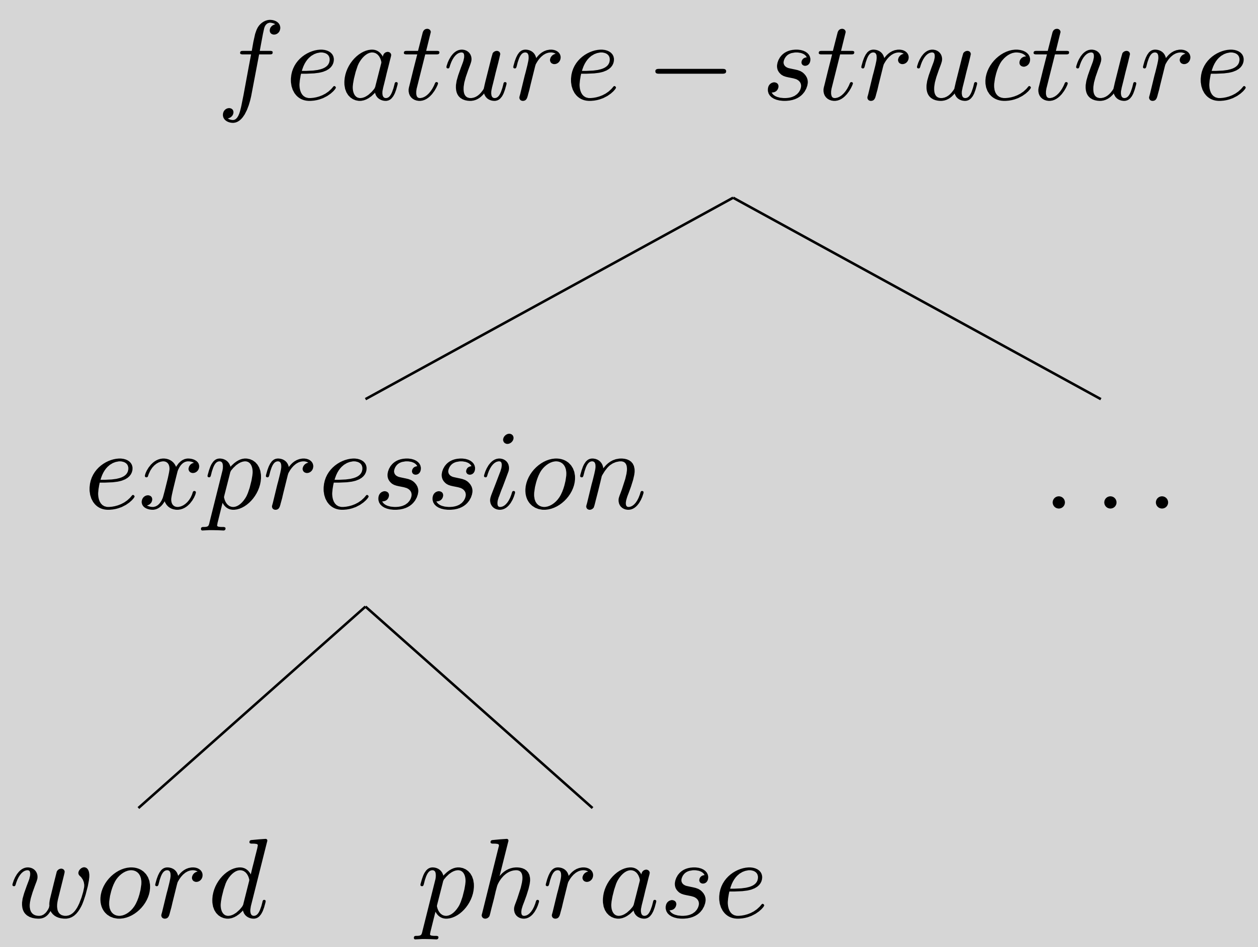
- The pizza example illustrates how unification can be used to combine information from different sources.
- In our grammar, information will come from lexical entries, grammar rules, and general principles.

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

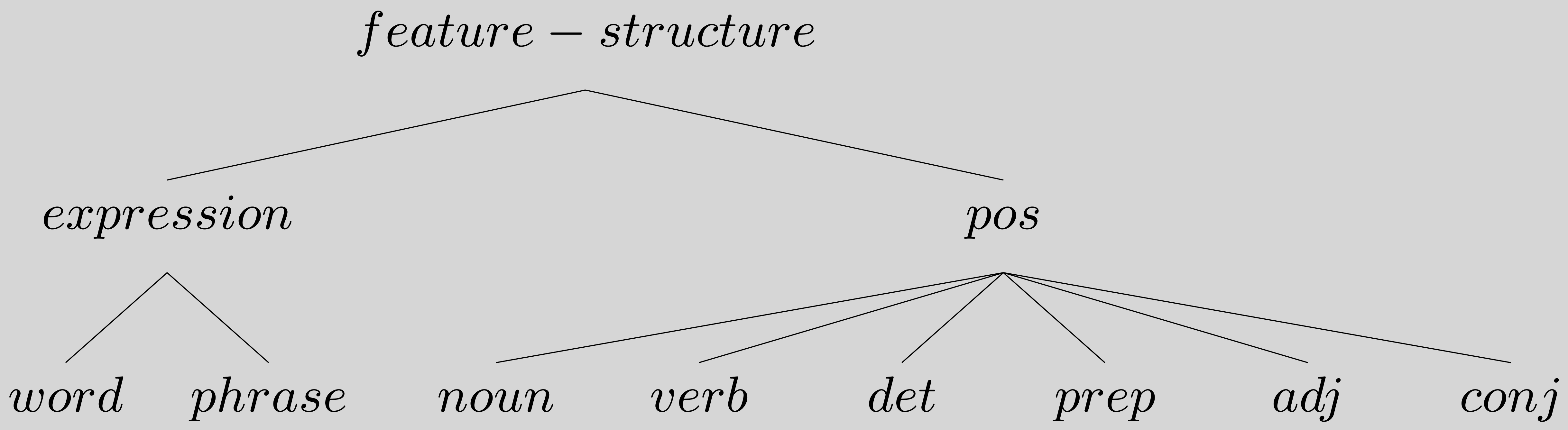


A Feature for Part of Speech

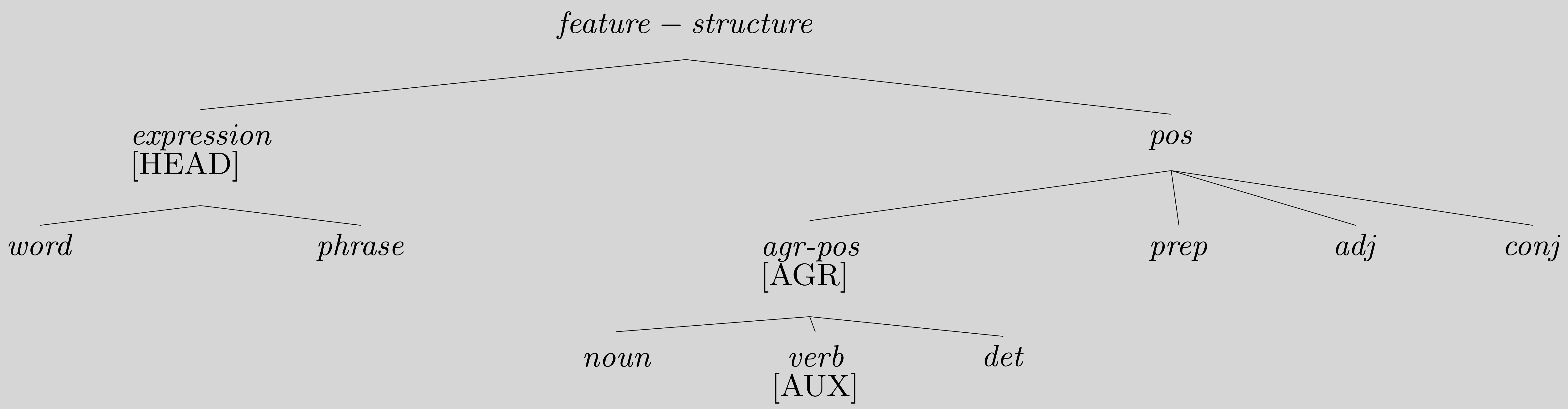
$$\text{NP} = \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{noun} \end{bmatrix}$$

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

Type Hierarchy for Parts of Speech I



Type Hierarchy for Parts of Speech II



A Feature for Valence

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

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

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

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

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

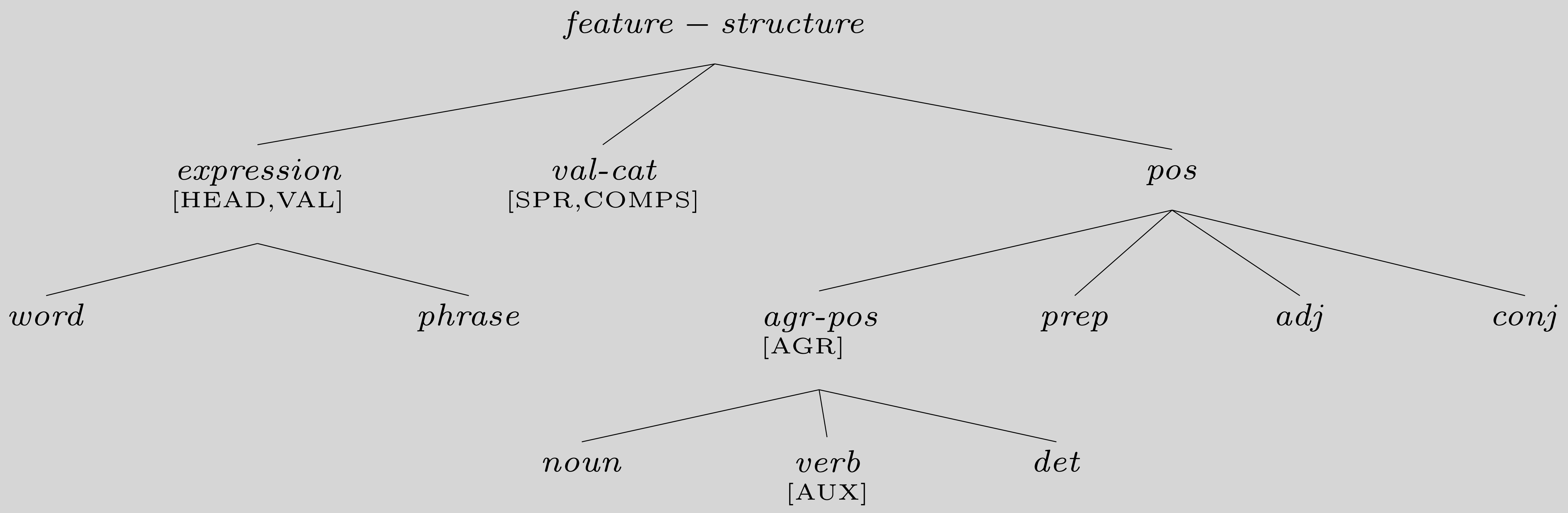
$$\text{VP} = \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

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:

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

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:

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

Head-Specifier Rule 2:

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

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

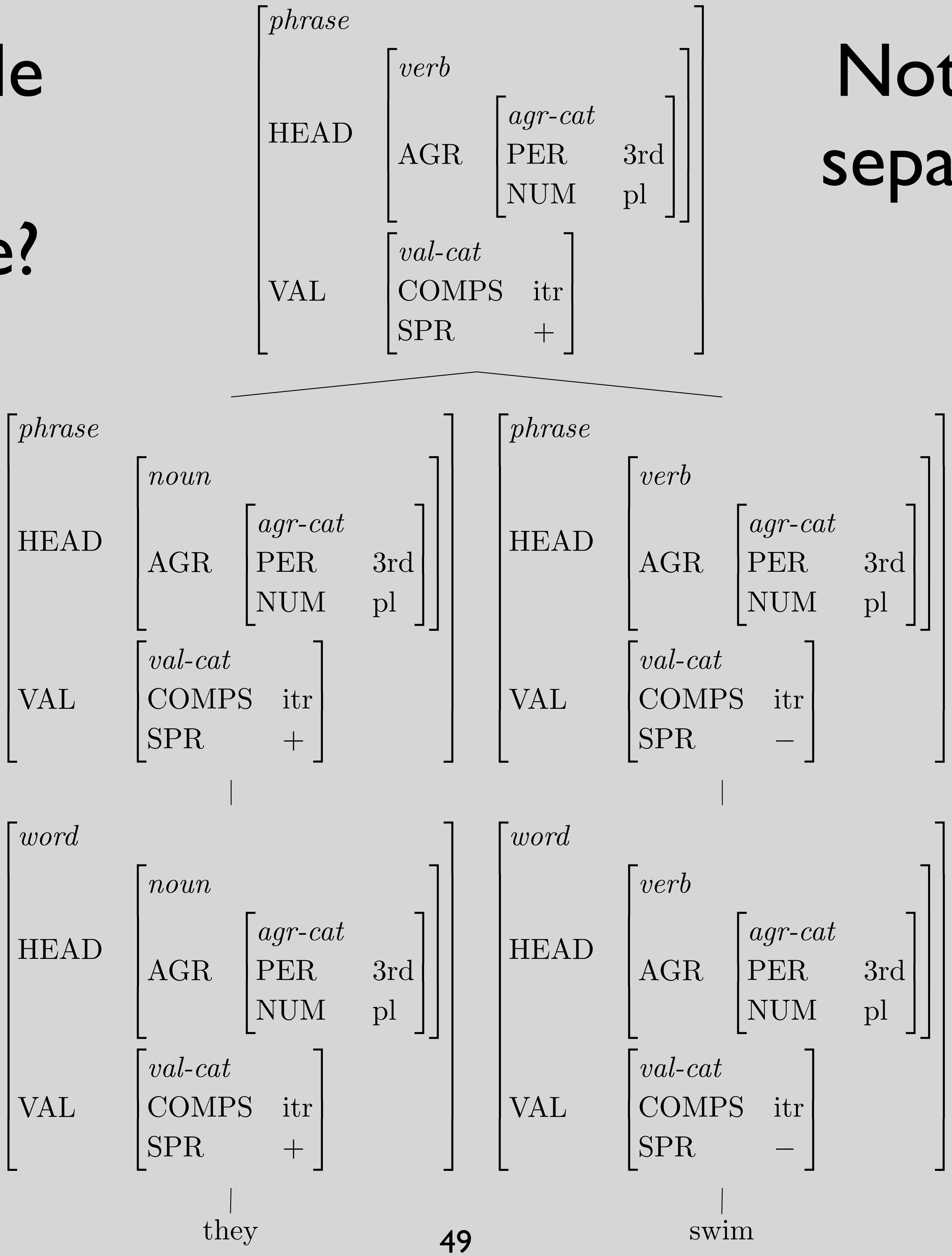
A Tree is Well-Formed if …

- 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?



phrase HEAD verb AGR agr-cat PER 3rd NUM pl VAL val-cat COMPS itr SPR +
phrase HEAD noun AGR agr-cat PER 3rd NUM pl VAL val-cat COMPS itr SPR +
phrase HEAD verb AGR agr-cat PER 3rd NUM pl VAL val-cat COMPS itr SPR −
word HEAD noun AGR agr-cat PER 3rd NUM pl VAL val-cat COMPS itr SPR +
word HEAD verb AGR agr-cat PER 3rd NUM pl VAL val-cat COMPS itr SPR −
they
swim

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?

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

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

- I thought only phrases had heads, do words have heads too? It looks like they both get the HEAD feature from expression (26), but won't the value of HEAD be determined by the word and not the phrase? Maybe I'm just getting caught up in the example too early...

Reading Questions

- What is the head value of these? Are they Ss?
  - (a) Where is Mufasa?
  - (b) In the valley of the wildebeest.

Reading Questions

- The primary purpose of HEAD is to ensure that the daughter parts of a tree agree with the larger structure of the tree right? If this is the case then any tree that cannot be made (because there is a disagreement somewhere between heads and daughters) cannot be licensed by the grammar right? Based on this, it would not be possible to have a tree that had multi headed phrases that disagree? aka VP and NP have to agree?

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:

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

Reading Questions

- Chapter 3 states that there is a specific type of phrases, which are headed. What will then be the examples of non-headed phrases? Isn't headedness a property of the language/grammar we are considering, so all entities of category phrase must be headed?

- The current coordination rule in Ch.3 does not specify a head (and is the only rule without a head). What would be the head of a coordination?

  - (1) Alice and Bob have a child.

  - (2) * Alice and Bob has a child.

- Both *Alice* and *Bob* are singular, but the coordination appears to be plural.

Reading Questions

- In discussing HFP, why do we avoid "attributing directionality of causation in the sharing of properties between phrases and their heads"?
- [62] gives an example of a tree that is not licensed by the grammar. I understand the concept behind HFP, but I don't see how it is violated through that tree. The mother head is a noun, and I am assuming the daughter head is a verb. How can we identify this by looking at the tree?

(62) A Tree Not Licensed by the Grammar

$$\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \textit{agr-cat} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \begin{bmatrix} \textit{agr-cat} & \\ \text{PER} & \text{2nd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{str} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \quad \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \textit{agr-cat} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix}$$

Reading Questions

- Why are nouns itr? Why are VPs itr?
- What would an NP with a complement look like?
- Why does the feature structure for determiner the in Alex denies the allegation (p. 71) have the COMPS feature specified as itr? Shouldn’t the want to combine with a complement (which in this case seems to be allegation)?
- I only thought of transitivity as a verb-y property, so I was surprised that things besides verbs were assigned a value for COMPS (itr). Why is this? Couldn't it be a property that's only appropriate for verbs?

Reading Questions

- Sometimes, such as in (50), the value for HEAD is a single part of speech tag like verb, and in that case if we write [HEAD   [1]verb] and then when we later use [1] somewhere else, it just refers to the single value of verb. But at other times, such as in (68), we're using a tag [1] to designate and refer to a whole feature-structure identity that itself also contains a different tag. It seems to me like there's a lot of flexibility here with how tags can be utilized to refer to particular linguistic constraints, depending on what level of precision we want to express and whether we want to leave some things intentionally underspecified. Is it actually the case that tags are meant to be flexible in how they can be applied? Or am I misunderstanding their usage?

Reading Questions

- What is the purpose/use of underspecification in relation to feature-structures which seem to rely completely on specification?
- When to write both NUM and PER for AGR and when is only one of them enough?
- I'm confused about the necessity of underspecification. I thought that the goal of branching away from CFG was to be able to support variety of structures that exist in languages without sacrificing the complexity of the formal grammar. How do we know when to underspecify?

Reading Questions

- Section 3.3.3 introduces pos, which is a subtype of feat-struc. I am wondering why it is called part-of-speech? and Why it is parallel to another subtype expressions? (It seems they are quite different things).
- Why are pos, expression, val-cat and agr-cat put in the same level in the type hierarchy tree? Shouldn't agr-cat be linked with AGR?
- What does it mean for types to be sister nodes?
- Where do strings and numbers fit in?

Reading Questions

- Is it like types in OOP?
- Was this done intentionally to make these objects entities easier to work with computationally , or was this simply the most natural way to model this phenomena?

Reading Questions

- "Head-Complement," "Head-Specifier," etc: How exactly do these relate to the rules from chapter 2? Are they direct conversions of rules such as "S -> NP VP" or are they more abstract and therefore capture more linguistic phenomena than the simplified rules from chapter 2?
- Is the feature structure a part of CFG, or they are totally different concept?

Reading Questions

- What are some more concrete applications of feature structures? Although the types and categories are really specific, at the same time it is still very abstract. Are they useful in the theoretical sense of improving our grammar and avoiding arbitrariness and other shortcomings of the CFGs of Chapter 2? Does this allow for clear explanations of why a sentence might be ungrammatical? I think modeling as a whole is difficult because we can only ever inch closer to a better description of the grammar, but how do we measure the relative success of different systems when there are always new sentences to consider?

Reading Questions

- The notion of features is a bit confusing to me because they feel arbitrarily flexible. Are there cases when it's useful to classify certain properties/features as optional or required? Is there a standardized format/library for programming sets of feature structures?

- The feature structures introduce a lot more complexity into the way we model sentences. In real-world systems, would such increased complexity be justified by performance gain?

Reading Questions

- Is all of this practical from a computational perspective? Is it necessary for an NLP system to be able to encode what our feature structure grammar encodes in order for it to perform properly?
- Features can help resolve ambiguity, but how would they affect bottom-up parsing overall?