

Ling 566

Oct 8, 2015

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.

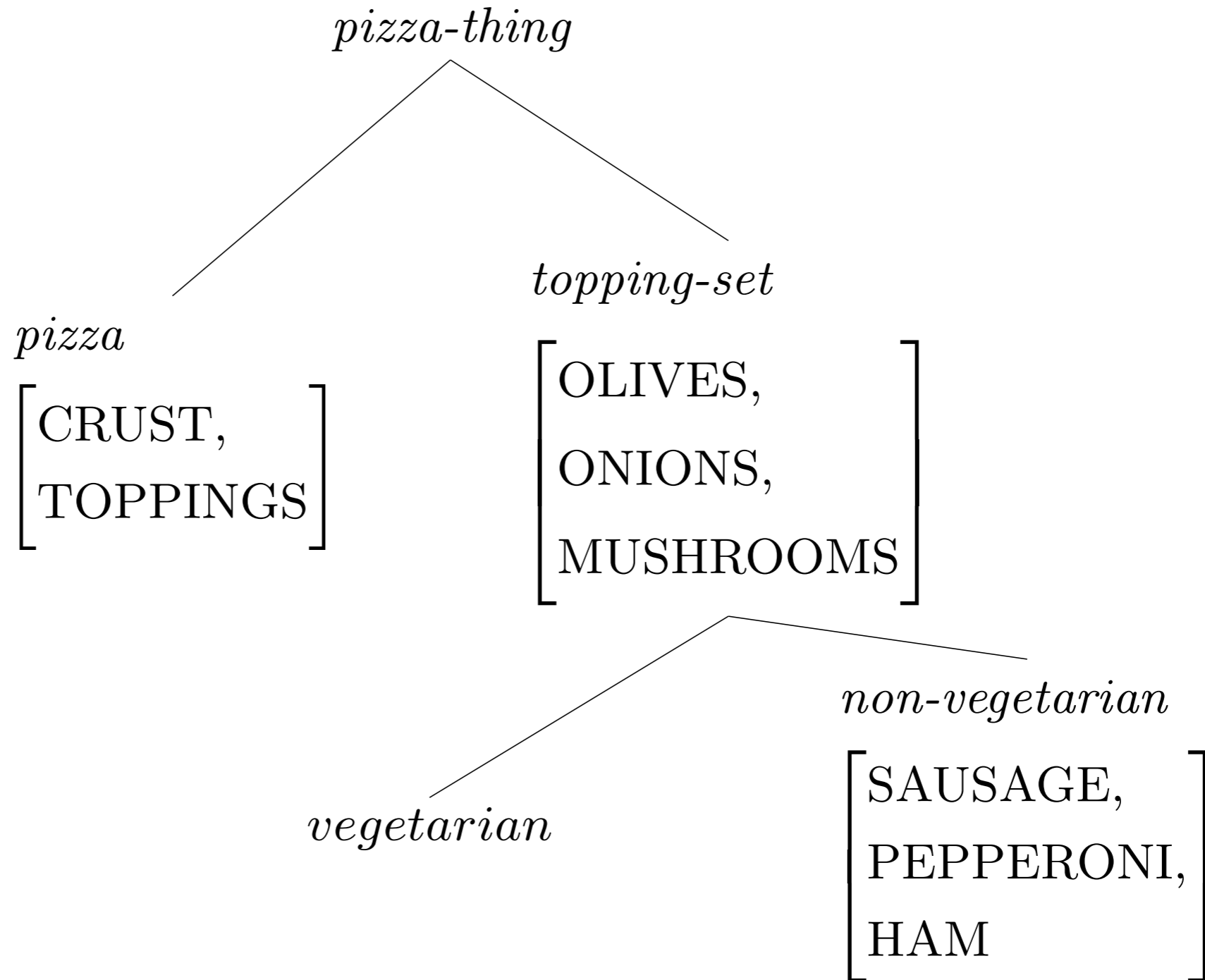
Cross-cutting Grammatical Properties

	3rd singular subject	plural subject
direct object NP	<i>denies</i>	<i>deny</i>
no direct object NP	<i>disappears</i>	<i>disappear</i>

Feature Structure Descriptions

FEATURE ₁	VALUE ₁
FEATURE ₂	VALUE ₂
...	
FEATURE _{<i>n</i>}	VALUE _{<i>n</i>}

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?

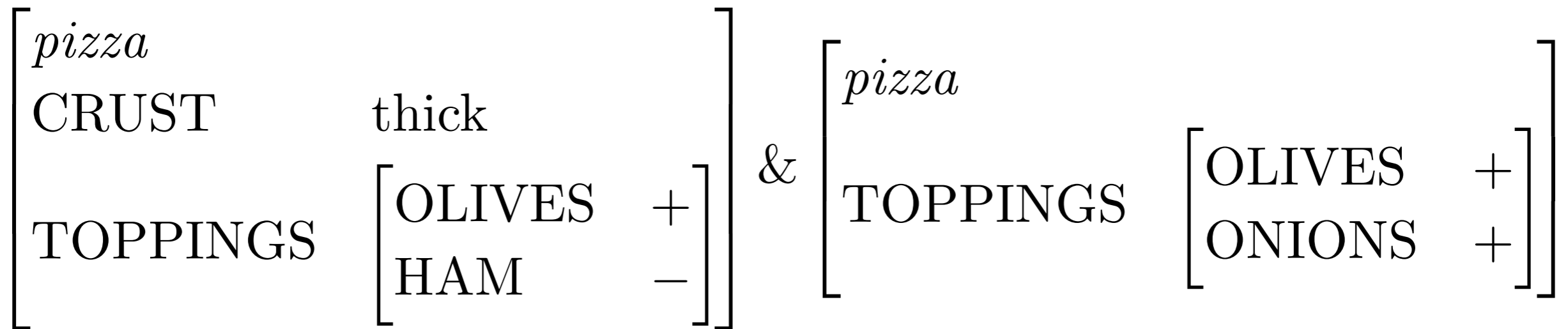
Answer: A large, constantly-changing number.

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



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

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

Combining Constraints

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

$$= \emptyset$$

Combining Constraints

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

$$= \emptyset$$

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

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

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

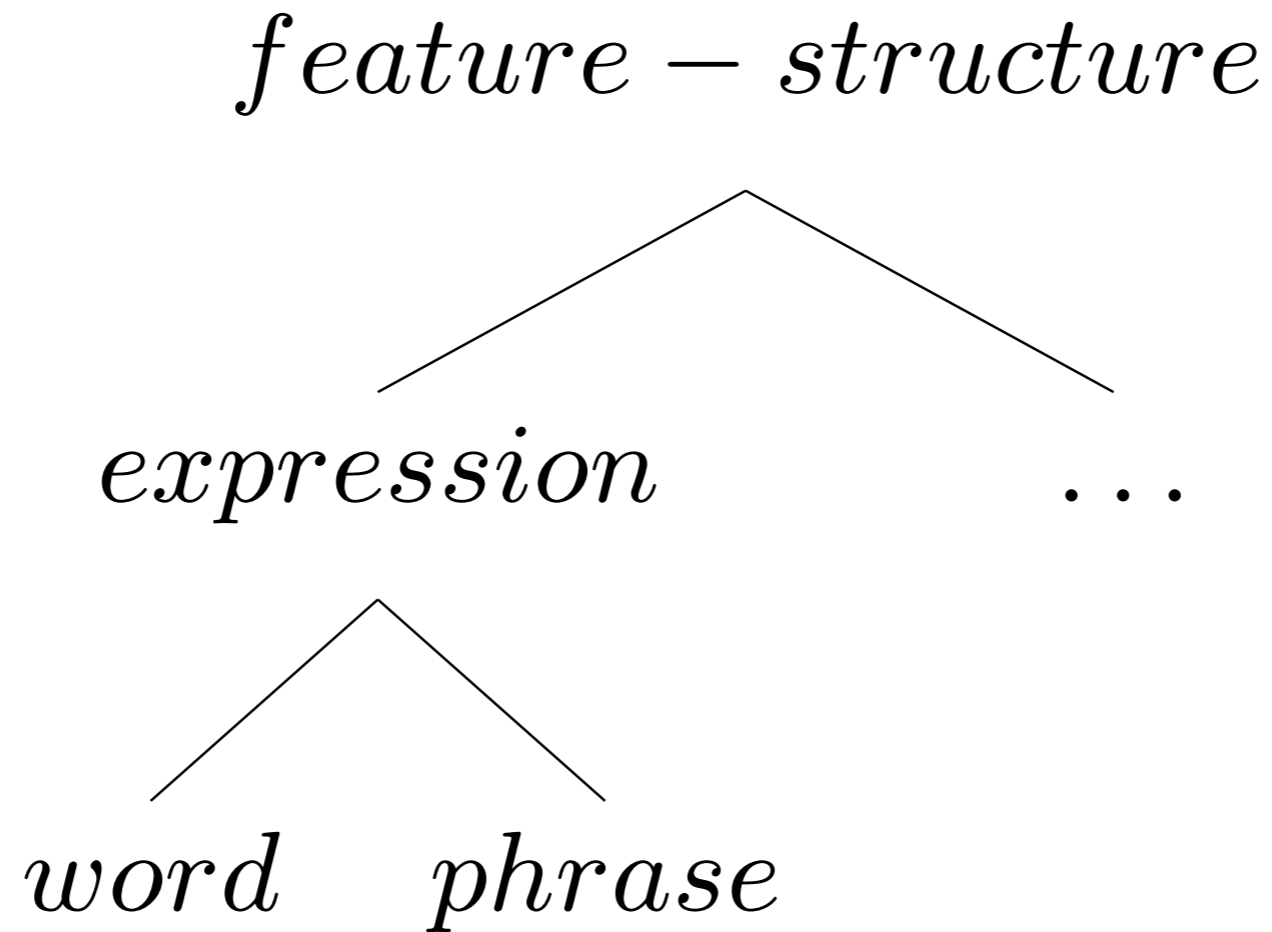
$$\left[\begin{array}{l} \textit{pizza} \\ \text{ONE-HALF} \\ \text{OTHER-HALF} \end{array} \right] \left[\begin{array}{l} \boxed{1} \left[\begin{array}{l} \text{ONIONS} \\ \text{OLIVES} \end{array} \right] \\ \boxed{1} \textit{vegetarian} \end{array} \right] \& \left[\begin{array}{l} \textit{pizza}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

$$= \emptyset$$

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.

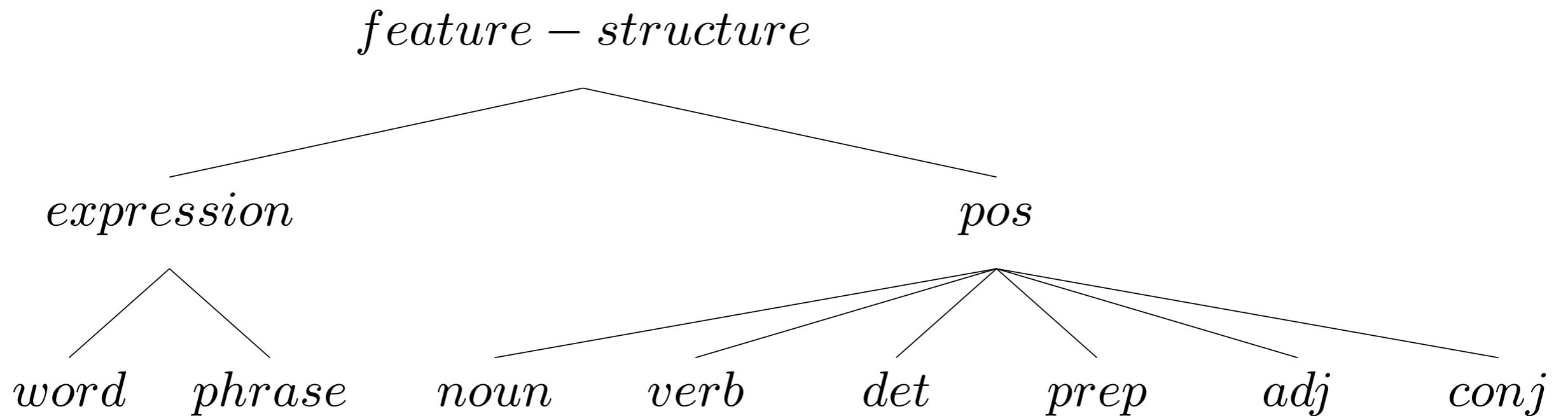
The Beginnings of Our Type Hierarchy



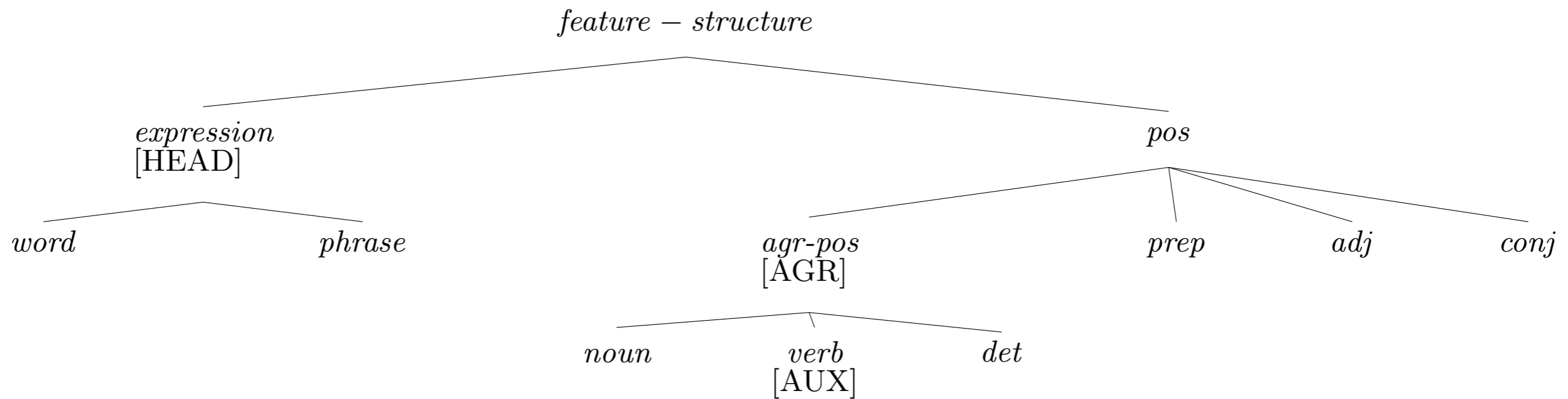
A Feature for Part of Speech

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

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

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

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

SPR and Verbs

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

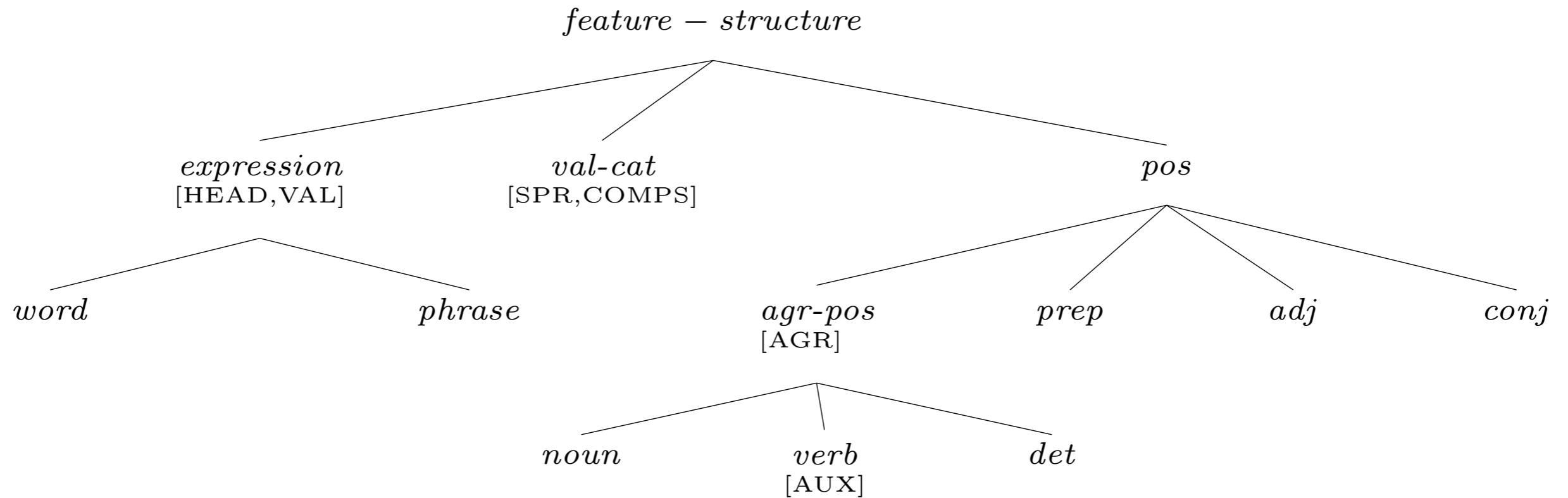
$$VP = \left[\begin{array}{l} \textit{phrase} \\ \text{HEAD} \quad \textit{verb} \\ \text{VAL} \quad \left[\begin{array}{l} \text{COMPS} \quad \textit{itr} \\ \text{SPR} \quad - \end{array} \right] \end{array} \right]$$

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:

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

Head Complement Rule 2:

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

Head Complement Rule 3:

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

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

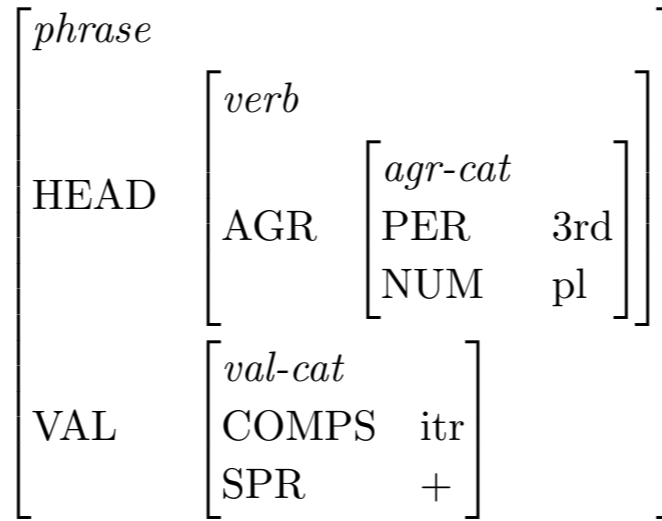
A Tree is Well-Formed if ...

- 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).

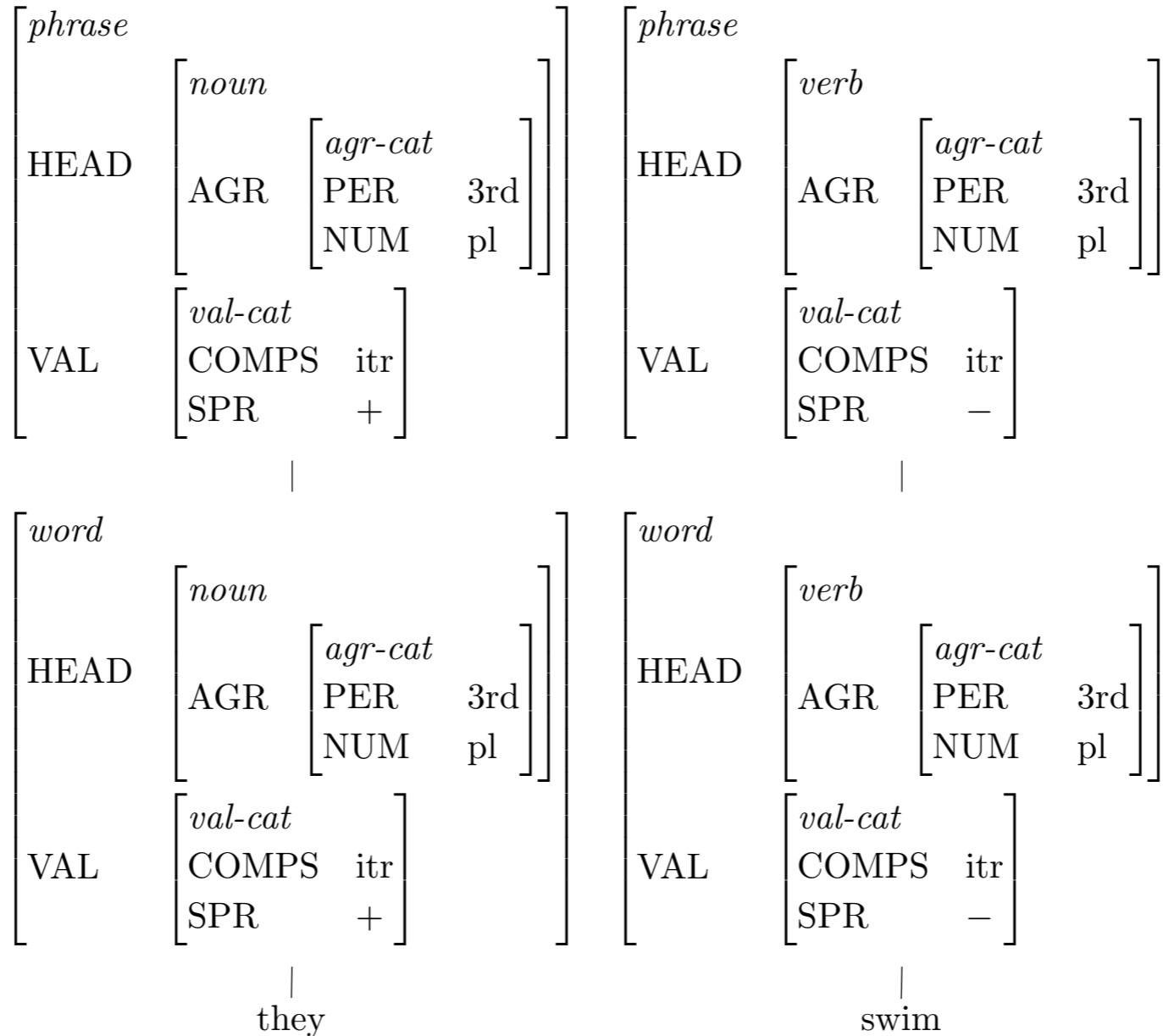
Question:

Do phrases that are not headed have
HEAD features?

Which rule licenses each node?

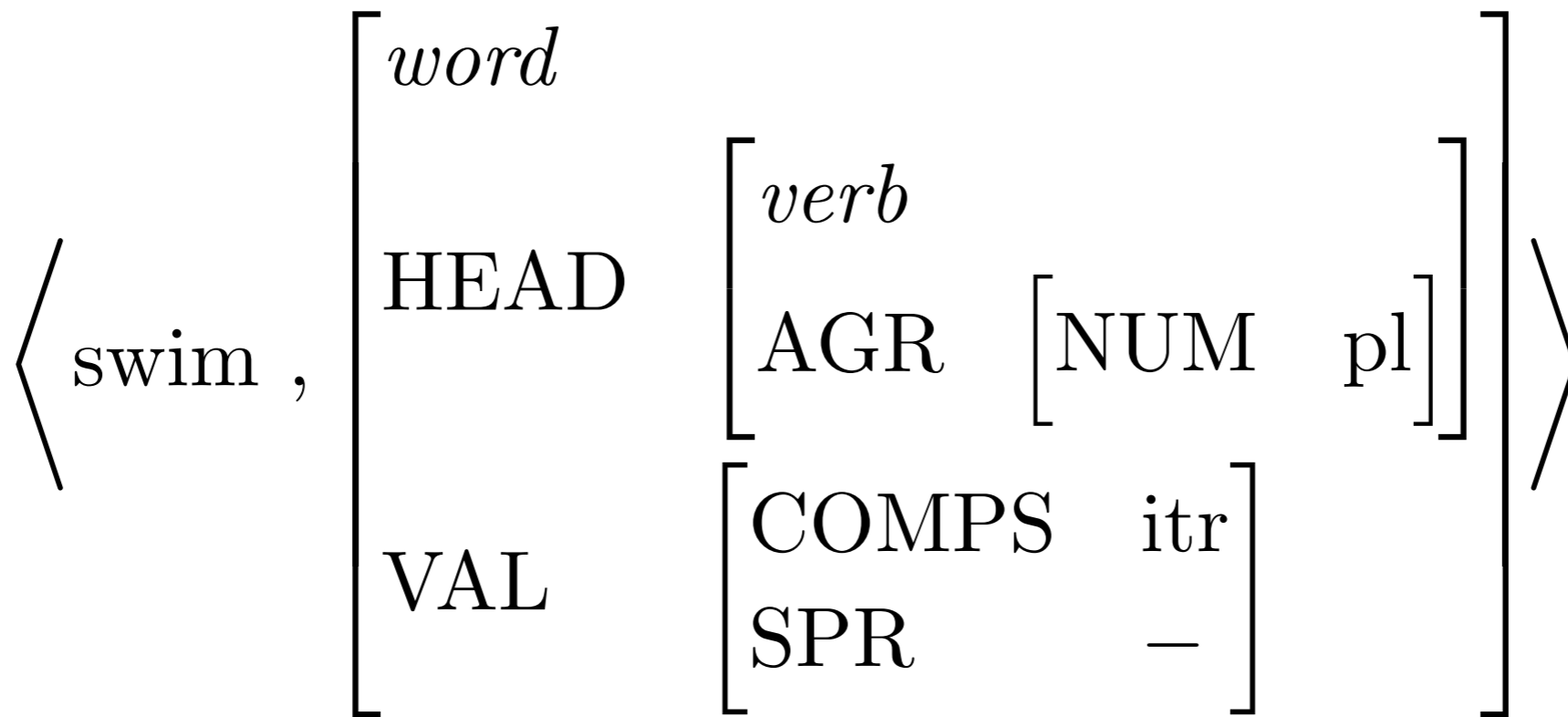


Note the three separate uses of DAGs



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 is word not a subtype of phrase in the type hierarchy? It seems that in all of the trees in the chapter, there is always a phrase at the root for a word so I would assume that word is a subtype of phrase.
- What is the purpose of *val-cat*? Why not just have SPR and COMPS as top-level features of expression?
- *pos* and *val-cat* are classed on the same level as *expression*. Is this because expression specifies the features HEAD and VAL whose values are these other types (or sub-types)? If so, then it seems to me that expression is in a sense dominating its sister nodes - why is this? And why then don't other sister types (e.g. the subtypes of *pos*) behave in the same way?

Reading Questions

- So, CFG's are insufficient because headedness is more than simply matching a V daughter to a VP mother. Headedness entails a bundle of features defined by the head daughter that the mother node must also have, including features associated with agreement of number and person. Are there more features encapsulated by the HEAD attribute, other than those in AGR, that we will come across later?
- I wonder when we determine a set of rules is too compact. As I was going through the HFP section, I was struck by the extent to which the headed rules had been simplified in (54). Are there disadvantages to removing HEAD value identities from a set of rules?

Reading Questions

- Why do we need to mark something with H in each rule?
- Why is the VP the head of S?

Reading Questions

- Why do VPs need a specifier? Isn't SPR for determiners?
- Why is VAL a feature of expression? Isn't valence only applicable to verbs?
- Are there any nouns that aren't [COMPS itr]?
- NOM is SPR -, NP is SPR +: Does NP always need a determiner? Shouldn't NP be able to function without existence of a determiner? And if so, shouldn't this rather be underspecified?

Reading Questions

- Why is the convention that nouns that have determiners are [SPR -] and nouns that never have determiners are [SPR +]. To me it more natural the other way around, i.e. a noun like "Alex" without a determiner has a negative value for SPR. Is the idea that a noun like "fact" has the negative value because on its own it lacks a determiner and needs to be assigned one?

Reading Questions

- The book talks about how both determiners and subject NP's are specifiers. Therefore, the relationship between subject and VP is parallel to determiner and NOM. This is because both subject NPs and determiners complete the phrases they are in, while NOM and VP are incomplete. I'm a little confused by this. I understand the parallel, but it seems counterintuitive to compare a subject NP to a determiners, as one is a phrase. What is gained by making this comparison?
- I guess one thing that threw me off a bit was, in the discussion about specifiers, the statement that NP is a fully formed category while VP is incomplete. Is this because the verb has been determined to have the role as head of a S and since S requires both VP and NP, we're basically saying that the onus is on the head to make sure all of the required components of its IST are present?

Reading Questions

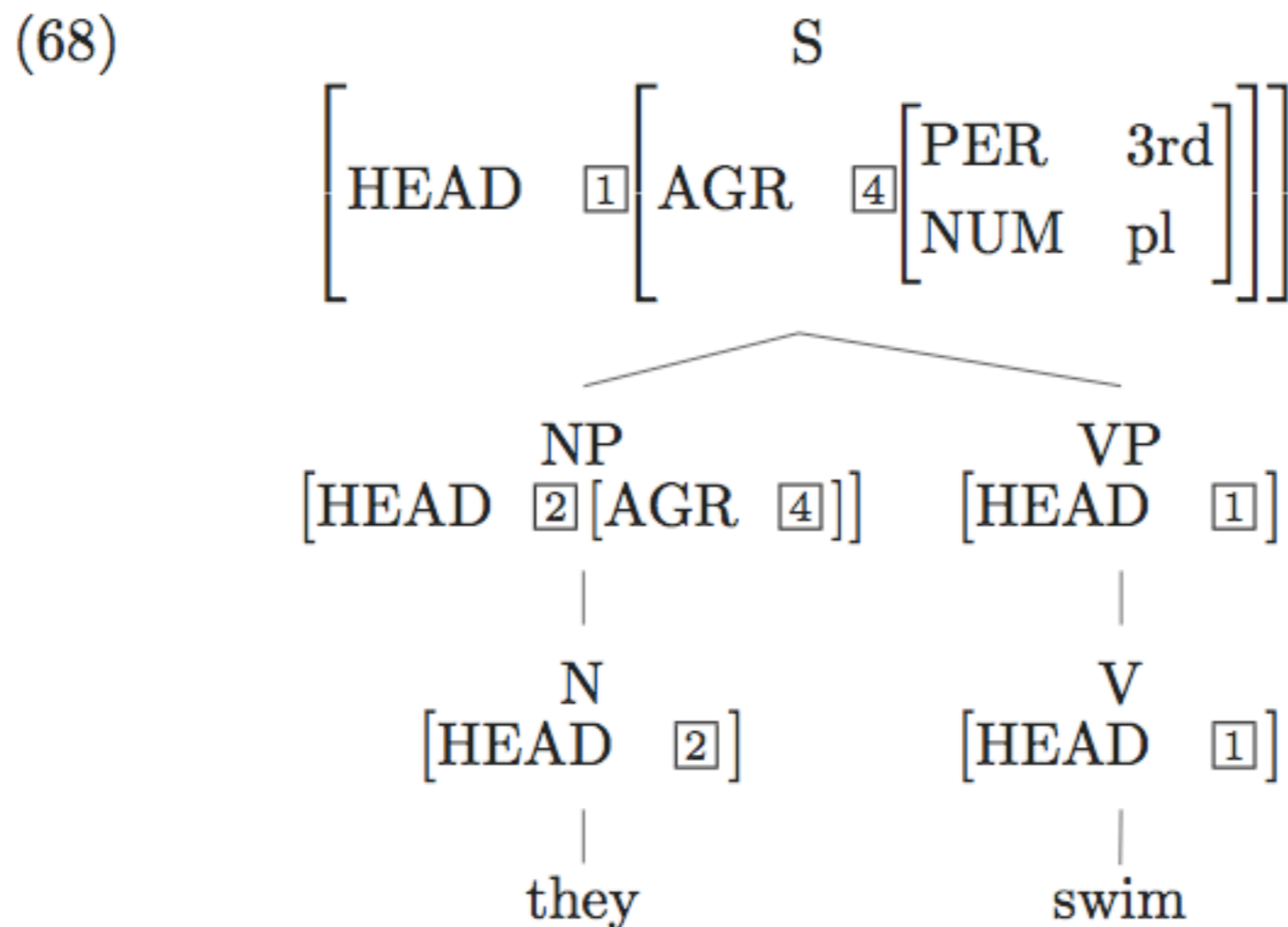
- How would one develop a grammar for phrase structures in free word order languages? Though most have preferred orders, creating rules for all possible orders would be rather a daunting proposition.
- How would phrase structure rules account for transposed word order in poetry or music (eg. "three sons had the king")?
- Are we just assuming there exists specific linear precedence rules for words or feature structures throughout this chapter? We have imposed principles such as HFP and a principle for valence. Would it be safe to assume that linear precedence is just a type of constraint?

Reading Questions

- When can feature specifications be optional?
- Can you have a structure for 3rd person pronoun analogous to underspecified "you", except instead of just the form "you", it is labeled with the set of words "he, she, it, they" then completed similarly by adding number. Or, considering a word with varying parts of speech such as "can"? Would the noun and auxiliary forms start as completely different lexical entries because they have a very different "sense"?
- The most intuitive approach for me to deal with ambiguous lexical terms would be to keep multiple entries, similar to the different senses in semantics. E.g. there would be separately kept entries for a 1st and 2nd person "you". What is the benefit of using an underspecified entry?

Reading Questions

- In (68) I'm not sure I understand the purpose of the HEAD and AGR tags. Do the different numbered tags resolve to the same thing?



Reading Questions

- Have all POS been known to show agreement in some language? Do certain POS pairs typically dominate in demonstrating agreement among the world's languages?
- Looking at just agreement structures, I'm wondering how feature structure diagrams differ between languages and what this means for applications. I know that in German, adjectives also need knowledge on agreement structures, as adjective endings change depending on case, plurality, and gender. Making a German feature structure, then, would require adjectives in the agr-pos category. It would also require different agreement features than English, which would, again, change the agreement structures heavily. This would change a lot of how the structure is formed, and I'm wondering how you would go about trying to translate- or another task- between two languages with, potentially, very different agreement structure formats.

Reading Questions

- What other feature-based grammars are out there? LFG seems to be a close cousin to HPSG; are there any good reasons for choosing one over the other (either for the purposes of doing linguistic theory or NLP)?
- Could we use feature structures to make a combined grammar/world knowledge model? How could such a thing be made probabilistic?
- Is the HPSG formalism like OOP? What are the parallels?