HG4041 Theories of Grammar

Feature Structures
Headed Rules, Trees

Francis Bond
Division of Linguistics and Multilingual Studies
http://www3.ntu.edu.sg/home/fcbond/
bond@ieee.org

Lecture 2
Location: LHN-TR+36

Overview

- Review: problems with CFG
- Modeling
- Feature structures, unification (pizza)
- Features for linguistic description
- Reformulate grammar rules
- Notion of head/headedness
- Licensing of trees

Our Goal s

- 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

- 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

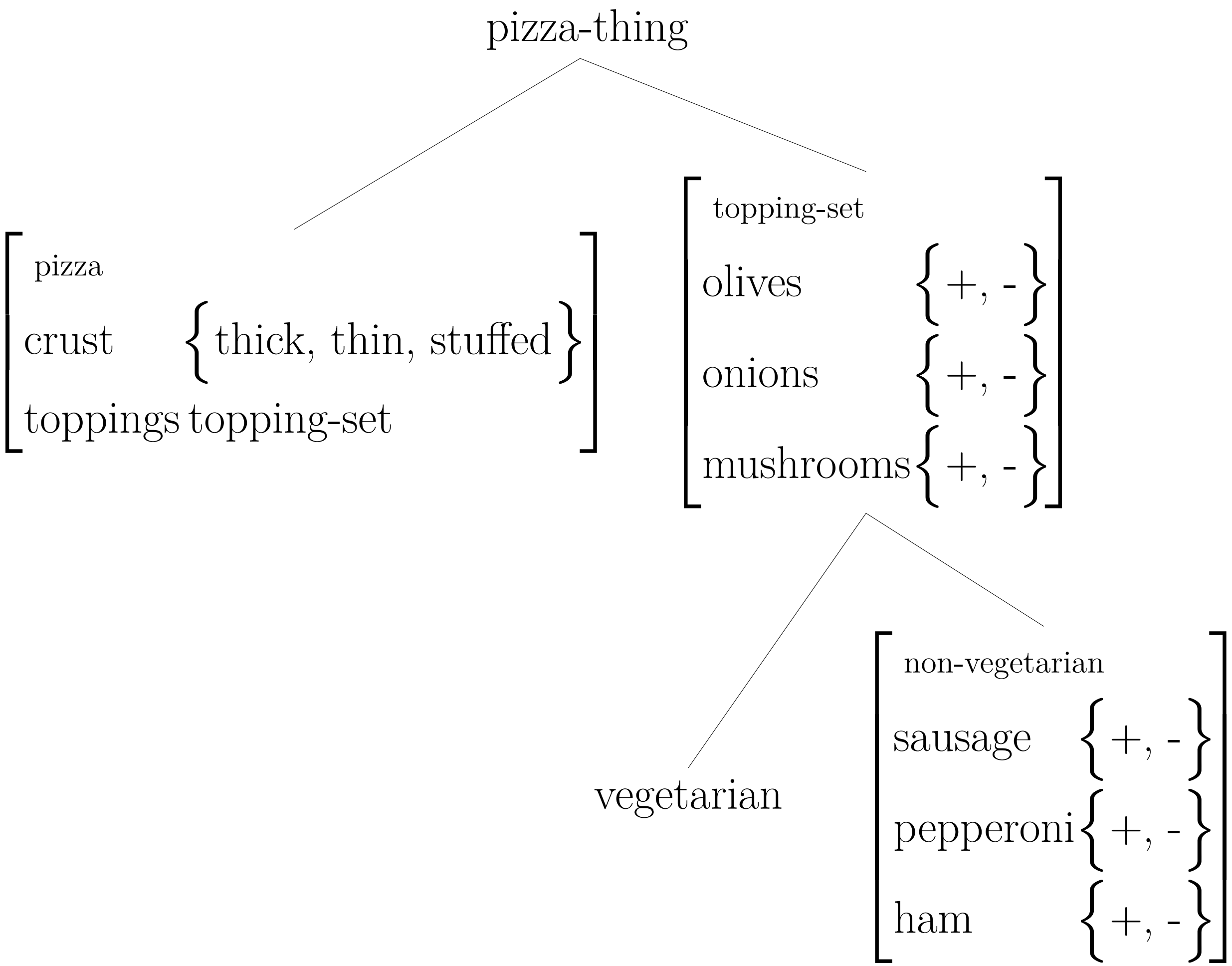
Language Models

- 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



pizza-thing

pizza
crust {thick, thin, stuffed}
toppings topping-set

topping-set
olives {+, -}
onions {+, -}
mushrooms {+, -}

vegetarian

non-vegetarian
sausage {+, -}
pepperoni {+, -}
ham {+, -}

Types

Type	Features/Values	IST
pizza-thing		NONE
pizza	$\begin{bmatrix} \text{crust} & \left\{\text{thick, thin, stuffed}\right\} \\ \text{toppings} & \text{topping-set} \end{bmatrix}$	pizza-thing
topping-set	$\begin{bmatrix} \text{olives} & \left\{+, -\right\} \\ \text{onions} & \left\{+, -\right\} \\ \text{mushrooms} & \left\{+, -\right\} \end{bmatrix}$	pizza-thing
vegetarian		topping-set
non-vegetarian	$\begin{bmatrix} \text{sausage} & \left\{+, -\right\} \\ \text{pepperoni} & \left\{+, -\right\} \\ \text{ham} & \left\{+, -\right\} \end{bmatrix}$	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

$$\begin{bmatrix} \text{pizza} & \\ \text{crust} & \text{thick} \\ \text{toppings} & \begin{bmatrix} \text{vegetarian} & \\ \text{olives} & + \\ \text{onions} & + \end{bmatrix} \end{bmatrix}$$

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

Answer: 2

$$\begin{bmatrix} {\scriptstyle \text{pizza}} & \\ \text{crust} & \text{thick} \\ \text{toppings} & \begin{bmatrix} {\scriptstyle \text{vegetarian}} & \\ \text{olives} & + \\ \text{onions} & + \\ \text{mushrooms} & \text{-} \end{bmatrix} \end{bmatrix} \begin{bmatrix} {\scriptstyle \text{pizza}} & \\ \text{crust} & \text{thick} \\ \text{toppings} & \begin{bmatrix} {\scriptstyle \text{vegetarian}} & \\ \text{olives} & + \\ \text{onions} & + \\ \text{mushrooms} & + \end{bmatrix} \end{bmatrix}$$

Pizza Descriptions and Pizza Models

$$\begin{bmatrix} \text{pizza} & \\ \text{crust} & \text{thick} \\ \text{toppings} & \begin{bmatrix} \text{vegetarian} & \\ \text{olives} & + \\ \text{onions} & + \end{bmatrix} \end{bmatrix}$$

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

A : A large, constantly-changing number.

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

Combining Constraints

$$\left[\begin{array}{ll} \text{pizza} & \\ \text{crust} & \text{thick} \\ \text{toppings} & \left[\begin{array}{ll} \text{olives} & + \\ \text{ham} & \text{-} \end{array}\right] \end{array}\right] \& \left[\begin{array}{ll} \text{pizza} & \\ \text{toppings} & \left[\begin{array}{ll} \text{olives} & + \\ \text{onions} & + \end{array}\right] \end{array}\right]$$

Unification is also written as: $\sqcap$.

Combining Constraints

$$\begin{bmatrix} \text{pizza} & \\ \text{crust} & \text{thick} \\ \text{toppings} & \begin{bmatrix} \text{olives} & + \\ \text{ham} & \text{-} \\ \text{onions} & + \end{bmatrix} \end{bmatrix}$$

Combining Constraints

$$\begin{bmatrix} {\scriptstyle \text{pizza}} & \\ \text{crust} & \text{thick} \\ \text{toppings} & \begin{bmatrix} \text{olives} & + \\ \text{ham} & \text{-} \end{bmatrix} \end{bmatrix} \& \begin{bmatrix} {\scriptstyle \text{pizza}} & \\ \text{crust} & \text{thin} \\ \text{toppings} & \begin{bmatrix} \text{olives} & + \\ \text{onions} & + \end{bmatrix} \end{bmatrix}$$

$$= \varphi$$

NULL is also written as: $\perp, \emptyset, \phi$.

Combining Constraints

Combining Constraints

$$\begin{bmatrix} \text{pizza} & \\ \text{crust} & \text{thin} \\ \text{toppings} & \begin{bmatrix} \text{olives} & + \\ \text{ham} & \text{-} \end{bmatrix} \end{bmatrix} \& \begin{bmatrix} \text{pizza} & \\ \text{crust} & \text{thin} \\ \text{toppings} & \begin{bmatrix} \text{vegetarian} \end{bmatrix} \end{bmatrix}$$

$$= \varphi$$

vegetarian has no feature ham.

A New Theory of Pizzas

$$\begin{bmatrix} {\scriptstyle \text{pizza}} & \\ \text{crust} & \left\{\text{thick, thin, stuffed}\right\} \\ \text{one-half} & \text{topping-set} \\ \text{other-half} & \text{topping-set} \end{bmatrix}$$

Combining Constraints

$$\begin{bmatrix} \text{pizza} & \\ \text{one-half} & \begin{bmatrix} \text{olives} & + \\ \text{ham} & - \end{bmatrix} \end{bmatrix} \& \begin{bmatrix} \text{pizza} & \\ \text{other-half} & \begin{bmatrix} \text{olives} & - \\ \text{ham} & + \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} \text{pizza} & \\ \text{one-half} & \begin{bmatrix} \text{olives} & + \\ \text{ham} & - \end{bmatrix} \\ \text{other-half} & \begin{bmatrix} \text{olives} & - \\ \text{ham} & + \end{bmatrix} \end{bmatrix}$$

Identity Constraints (tags)

$$\begin{bmatrix} \text{pizza} & \\ \text{one-half} & \begin{bmatrix} \text{olives} & \boxed{1} \\ \text{ham} & \boxed{2} \end{bmatrix} \\ \text{other-half} & \begin{bmatrix} \text{olives} & \boxed{1} \\ \text{ham} & \boxed{2} \end{bmatrix} \end{bmatrix}$$

Combining Constraints

$$\begin{bmatrix} \text{pizza} & & \\ \text{one-half} & \boxed{1} & \begin{bmatrix} \text{olives} & + \\ \text{ham} & \text{-} \end{bmatrix} \\ \text{other-half} & \boxed{1} & \end{bmatrix} \& \begin{bmatrix} \text{pizza} & \\ \text{other-half} & \begin{bmatrix} \text{olives} & + \\ \text{mushroom} & + \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} \text{pizza} & & \\ \text{one-half} & \boxed{1} & \begin{bmatrix} \text{olives} & + \\ \text{ham} & \text{-} \\ \text{mushroom} & + \end{bmatrix} \\ \text{other-half} & \boxed{1} & \end{bmatrix}$$

Note

$$\begin{bmatrix} \text{pizza} & \\ \text{one-half} & \boxed{1} \begin{bmatrix} \text{olives} & + \\ \text{ham} & \text{-} \\ \text{mushroom} & + \end{bmatrix} \\ \text{other-half} & \boxed{1} \end{bmatrix}$$

$$= \begin{bmatrix} \text{pizza} & \\ \text{one-half} & \boxed{1} \\ \text{other-half} & \boxed{1} \begin{bmatrix} \text{olives} & + \\ \text{ham} & \text{-} \\ \text{mushroom} & + \end{bmatrix} \end{bmatrix}$$

Combining Constraints

$$\begin{bmatrix} \text{\scriptsize pizza} & \\ \text{crust} & \text{thick} \\ \text{one-half} & \boxed{1}\begin{bmatrix} \text{olives} & + \\ \text{onion} & \text{-} \end{bmatrix} \\ \text{other-half} & \boxed{1}\ \text{vegetarian} \end{bmatrix} \& \begin{bmatrix} \text{\scriptsize pizza} & \\ \text{crust} & \text{thin} \\ \text{one-half} & \begin{bmatrix} \text{olives} & + \\ \text{pepperoni} & + \end{bmatrix} \end{bmatrix}$$

$$= \varphi$$

Why combine constraints?

- The pizza example illustrates how unification can be used to combine information from different sources.

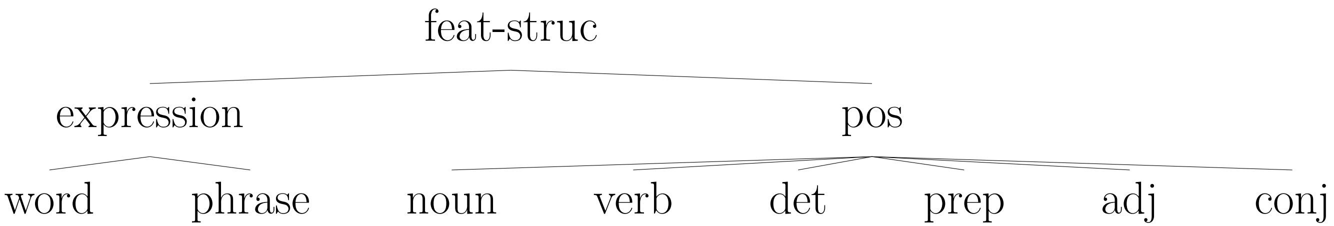
- In our grammar, information will come from lexical entries, grammar rules, and general principles.

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



feat-struc

expression

pos

word

phrase

noun

verb

det

prep

adj

conj

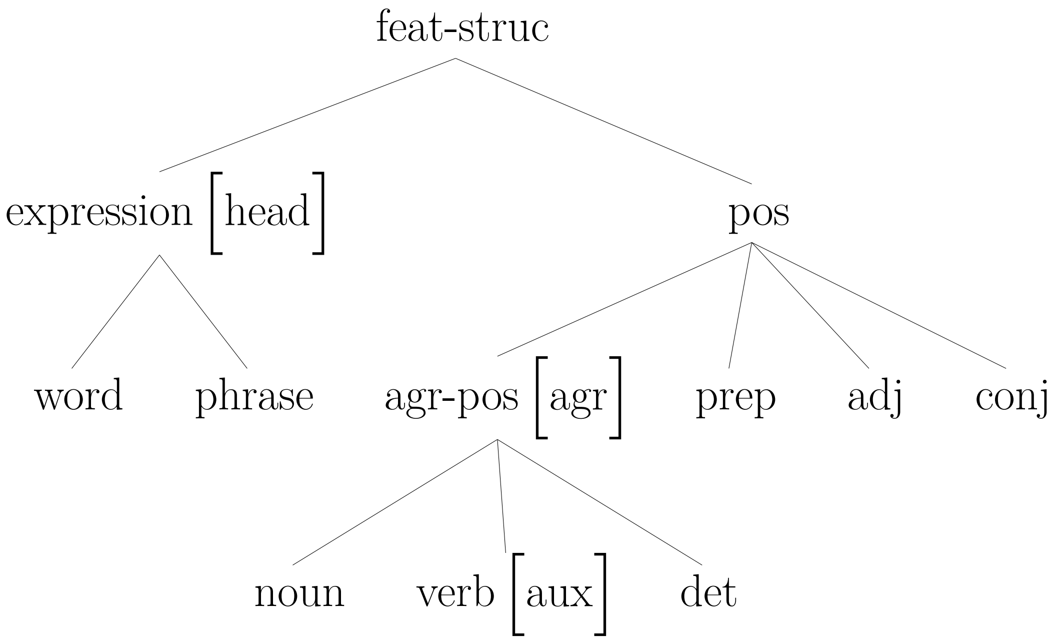
NP $\left[\begin{array}{ll}\textit{phrase} & \\ \text{head} & \text{noun}\end{array}\right]$

VP $\left[\begin{array}{ll}\textit{phrase} & \\ \text{head} & \text{verb}\end{array}\right]$

N $\left\langle \text{fly} \; , \left[\begin{array}{ll}\textit{word} & \\ \text{head} & \text{noun}\end{array}\right]\right\rangle$

V $\left\langle \text{fly} \; , \left[\begin{array}{ll}\textit{word} & \\ \text{head} & \text{verb}\end{array}\right]\right\rangle$

Type Hierarchy for Parts of Speech II



Agreement

We need more information to make words agree.

$$\left\langle \text{fly}, \begin{bmatrix} \text{word} & & \\ \text{head} & \text{noun} & \begin{bmatrix} \text{agr-cat} & \\ \text{agr} & \begin{bmatrix} \text{per} & 3 \\ \text{num} & \text{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{flies}, \begin{bmatrix} \text{word} & & \\ \text{head} & \text{verb} & \begin{bmatrix} \text{agr-cat} & \\ \text{agr} & \begin{bmatrix} \text{per} & 3 \\ \text{num} & \text{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle$$

Agreement

$$\begin{bmatrix} \text{phrase} & \\ \text{head} & \boxed{1}\text{verb} \\ \text{val} & \begin{bmatrix} \text{comps} & \text{itr} \\ \text{spr} & + \end{bmatrix} \end{bmatrix} \rightarrow \overset{\text{NP}}{\begin{bmatrix} \text{head} & \begin{bmatrix} \text{agr} & \boxed{2} \end{bmatrix} \end{bmatrix}} \begin{bmatrix} \text{phrase} & \\ \text{head} & \boxed{1}\begin{bmatrix} \text{agr} & \boxed{2} \end{bmatrix} \\ \text{val} & \begin{bmatrix} \text{comps} & \text{itr} \\ \text{spr} & - \end{bmatrix} \end{bmatrix}$$

The values on agr for the subject NP and verb phrase must be identical.

A Simple Feature for Valence

$$\text{iv} = \begin{bmatrix} \text{word} & \\ \text{head} & \text{verb} \\ \text{val} & \begin{bmatrix} \text{val-cat} & \\ \text{comps} & \text{itr} \end{bmatrix} \end{bmatrix} \qquad \text{tv} = \begin{bmatrix} \text{word} & \\ \text{head} & \text{verb} \\ \text{val} & \begin{bmatrix} \text{val-cat} & \\ \text{comps} & \text{str} \end{bmatrix} \end{bmatrix}$$

$$\text{dtv} = \begin{bmatrix} \text{word} & \\ \text{head} & \text{verb} \\ \text{val} & \begin{bmatrix} \text{val-cat} & \\ \text{comps} & \text{dtr} \end{bmatrix} \end{bmatrix}$$

comps controls how many complements are possible.

Head-Complement Rules

$$1 \quad \begin{bmatrix} \text{phrase} & \\ \text{head} & \boxed{1} \\ \text{val} & \begin{bmatrix} \text{comps} & \text{itr} \\ \text{spr} & - \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \text{word} & \\ \text{head} & \boxed{1} \\ \text{val} & \begin{bmatrix} \text{comps} & \text{itr} \\ \text{spr} & - \end{bmatrix} \end{bmatrix}$$

$$2 \quad \begin{bmatrix} \text{phrase} & \\ \text{head} & \boxed{1} \\ \text{val} & \begin{bmatrix} \text{comps} & \text{itr} \\ \text{spr} & - \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \text{word} & \\ \text{head} & \boxed{1} \\ \text{val} & \begin{bmatrix} \text{comps} & \text{str} \\ \text{spr} & - \end{bmatrix} \end{bmatrix} \ \text{NP}$$

$$3 \quad \begin{bmatrix} \text{phrase} & \\ \text{head} & \boxed{1} \\ \text{val} & \begin{bmatrix} \text{comps} & \text{itr} \\ \text{spr} & - \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \text{word} & \\ \text{head} & \boxed{1} \\ \text{val} & \begin{bmatrix} \text{comps} & \text{dtr} \\ \text{spr} & - \end{bmatrix} \end{bmatrix} \ \text{NP} \ \ \text{NP}$$

Underspecification

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

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

$$\begin{bmatrix} \text{head} & \text{verb} \end{bmatrix}$$

Another Valence Feature

$$\text{NP} = \begin{bmatrix} \text{phrase} & \\ \text{head} & \text{noun} \\ \text{val} & \begin{bmatrix} \text{val-cat} & \\ \text{comps} & \text{itr} \\ \text{spr} & + \end{bmatrix} \end{bmatrix} \qquad \text{NOM} = \begin{bmatrix} \text{phrase} & \\ \text{head} & \text{noun} \\ \text{val} & \begin{bmatrix} \text{val-cat} & \\ \text{comps} & \text{itr} \\ \text{spr} & - \end{bmatrix} \end{bmatrix}$$

spr controls the specifier (determiner and/or subject)

spr and Verbs

$$\mathrm{S} = \begin{bmatrix} \text{phrase} & \\ \text{head} & \text{verb} \\ \text{val} & \begin{bmatrix} \text{val-cat} & \\ \text{comps} & \text{itr} \\ \text{spr} & + \end{bmatrix} \end{bmatrix} \qquad \mathrm{VP} = \begin{bmatrix} \text{phrase} & \\ \text{head} & \text{verb} \\ \text{val} & \begin{bmatrix} \text{val-cat} & \\ \text{comps} & \text{itr} \\ \text{spr} & - \end{bmatrix} \end{bmatrix}$$

S and NP

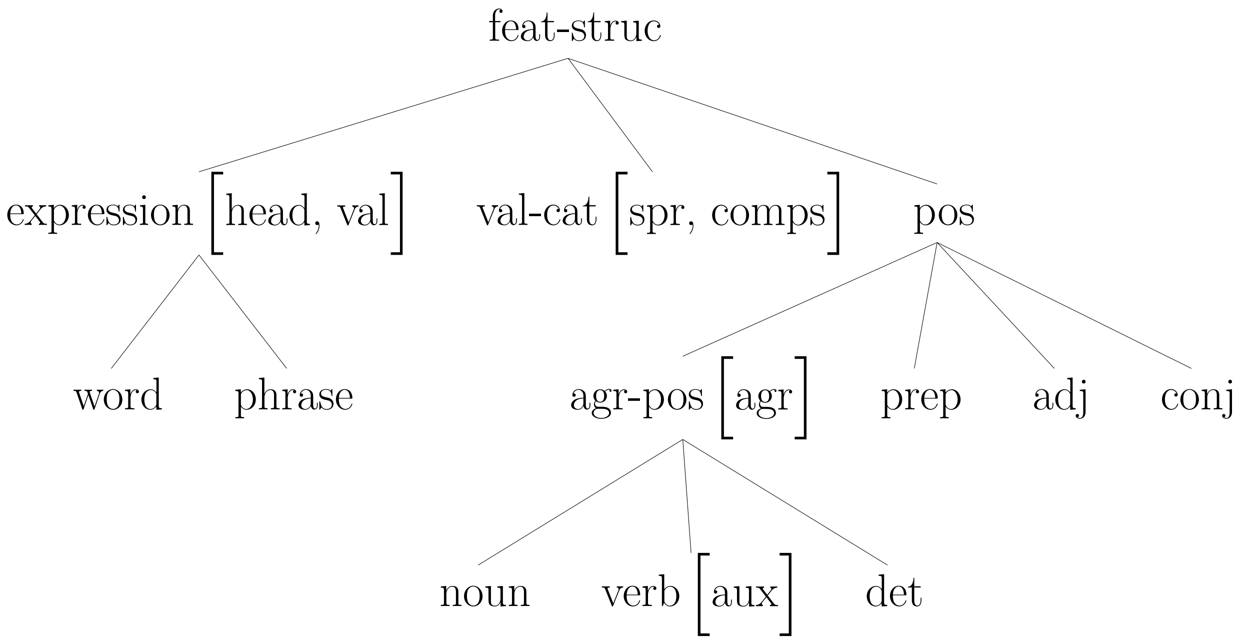
$$\begin{bmatrix} \text{val} & \begin{bmatrix} \text{val-cat} & \\ \text{comps} & \text{itr} \\ \text{spr} & + \end{bmatrix} \end{bmatrix}$$

- both are fully saturated: specified and no more complements

(1) We created a monster

(2) our creation of a monster

Type Hierarchy So Far



Heads

- Intuitive idea: A phrase typically contains a word that determines its most essential properties, including
  - where it occurs in larger phrases
  - 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
  can you think of a phrase that doesn't?

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

- the head daughter in a headed-rule will be labeled with 'H'.

  [type] → ... H [ ] ...

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?

Reformulating the Grammar Rules I

Which Ch 2 rules do these correspond to?

- Head-Complement Rule 1:

$$\begin{bmatrix}\text{phrase} & \\ \text{val} & \begin{bmatrix}\text{comps} & \text{itr}\\ \text{spr} & -\end{bmatrix}\end{bmatrix} \rightarrow \text{H}\begin{bmatrix}\text{word} & \\ \text{val} & \begin{bmatrix}\text{comps} & \text{itr}\\ \text{spr} & -\end{bmatrix}\end{bmatrix}$$

- Head-Complement Rule 2:

$$\begin{bmatrix}\text{phrase} & \\ \text{val} & \begin{bmatrix}\text{comps} & \text{itr}\\ \text{spr} & -\end{bmatrix}\end{bmatrix} \rightarrow \text{H}\begin{bmatrix}\text{word} & \\ \text{val} & \begin{bmatrix}\text{comps} & \text{str}\\ \text{spr} & -\end{bmatrix}\end{bmatrix} \ \text{NP}$$

- Head-Complement Rule 3:

$$\begin{bmatrix} \text{phrase} & \\ \text{val} & \begin{bmatrix} \text{comps} & \text{itr} \\ \text{spr} & - \end{bmatrix} \end{bmatrix} \rightarrow \text{H} \begin{bmatrix} \text{word} & \\ \text{head} & \boxed{1} \\ \text{val} & \begin{bmatrix} \text{comps} & \text{dtr} \\ \text{spr} & - \end{bmatrix} \end{bmatrix} \ \text{NP} \quad \text{NP}$$

Reformulating the Grammar Rules II

- Head-Specifier Rule 1:

$$\begin{bmatrix} \text{phrase} & \\ \text{val} & \begin{bmatrix} \text{comps} & \text{itr} \\ \text{spr} & + \end{bmatrix} \end{bmatrix} \rightarrow \overset{\text{NP}}{\begin{bmatrix} \text{head} & \begin{bmatrix} \text{agr} & \boxed{1} \end{bmatrix} \end{bmatrix}} \;\; \text{H}\begin{bmatrix} \text{phrase} & \\ \text{head} & \begin{bmatrix} \text{verb} & \\ \text{agr} & \boxed{1} \end{bmatrix} \\ \text{val} & \begin{bmatrix} \text{spr} & - \end{bmatrix} \end{bmatrix}$$

- Head-Specifier Rule 2:

$$\begin{bmatrix} \text{phrase} & \\ \text{val} & \begin{bmatrix} \text{comps} & \text{itr} \\ \text{spr} & + \end{bmatrix} \end{bmatrix} \rightarrow \text{D} \quad \text{H}\begin{bmatrix} \text{phrase} & \\ \text{head} & \text{noun} \\ \text{val} & \begin{bmatrix} \text{spr} & - \end{bmatrix} \end{bmatrix}$$

Reformulating the Grammar Rules III

- Non-Branching NP Rule

$$\begin{bmatrix} \text{phrase} & \\ \text{val} & \begin{bmatrix} \text{comps} & \text{itr} \\ \text{spr} & + \end{bmatrix} \end{bmatrix} \rightarrow \text{H}\begin{bmatrix} \text{word} & \\ \text{head} & \text{noun} \\ \text{val} & \begin{bmatrix} \text{spr} & + \end{bmatrix} \end{bmatrix}$$

- Head-Modifier Rule

$$\begin{bmatrix} \text{phrase} & \\ \text{val} & \begin{bmatrix} \text{comps} & \text{itr} \\ \text{spr} & - \end{bmatrix} \end{bmatrix} \rightarrow \text{H}\begin{bmatrix} \text{phrase} & \\ \text{val} & \begin{bmatrix} \text{spr} & - \end{bmatrix} \end{bmatrix} \text{PP}$$

- Coordination Rule

$$\begin{bmatrix}\text{head} & \boxed{1}\end{bmatrix} \rightarrow \begin{bmatrix}\text{head} & \boxed{1}\end{bmatrix} + \begin{bmatrix}\text{word} & \\ \text{head} & \text{conj}\end{bmatrix} \begin{bmatrix}\text{head} & \boxed{1}\end{bmatrix}$$

Only coordinate like things!

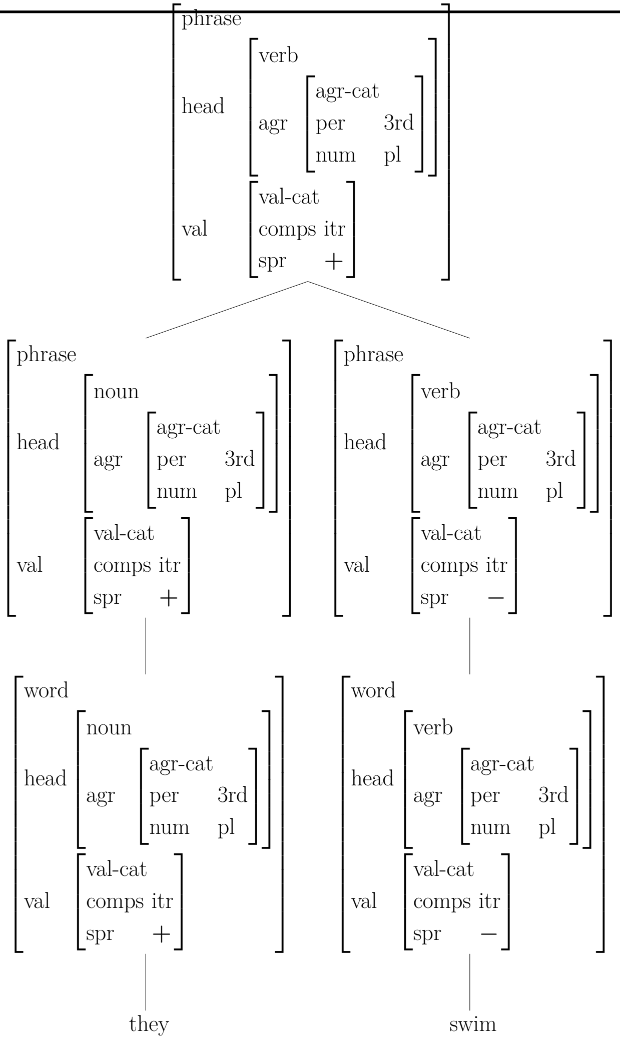
Advantages of the New Formulation

- 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
- Anything else?

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 −

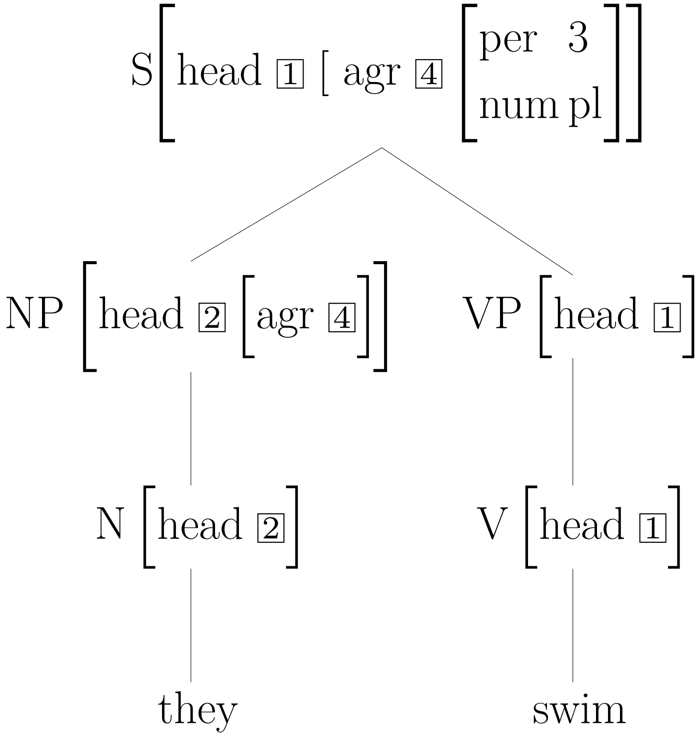
word
head
noun
agr
agr-cat
per 3rd
num pl
val
val-cat
comps itr
spr +

word
head
verb
agr
agr-cat
per 3rd
num pl
val
val-cat
comps itr
spr −

they

swim

In abbreviated form



S ⇒ phrase, head verb, val itr, spr val+;
VP ⇒ phrase, head verb, val itr, spr val-; ...

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?

$$\left\langle \text{swim}, \begin{bmatrix} \text{word} & \\ \text{head} & \begin{bmatrix} \text{verb} & \\ \text{agr} & \begin{bmatrix} \text{num} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{val} & \begin{bmatrix} \text{val-cat} & \\ \text{comps} & \text{itr} \\ \text{spr} & \text{-} \end{bmatrix} \end{bmatrix} \right\rangle$$

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: complex feature values

Acknowledgments and References

- Course design and slides borrow heavily from Emily Bender's course: Linguistics 566: Introduction to Syntax for Computational Linguistics http://courses.washington.edu/ling566