Ling/CSE 472: Introduction to Computational Linguistics

5/13/15

Feature structures and unification

Overview

- Problems with CFG
- Feature structures
- Unification
- Agreement
- Subcategorization
- Long-distance Dependencies
- Reading questions

Problems with CFG (with atomic node labels)

- Simple rules, with simple category sets overgenerate:

- What are some non-sentences that this CFG licenses?

$$
\begin{array}{ll}
\text{S} \rightarrow & \text{NP VP} \\
\text{NP} \rightarrow & \text{(Det) Noun} \\
\text{VP} \rightarrow & \text{Verb (NP) (NP|PP)} \\
\text{PP} \rightarrow & \text{Prep NP} \\
\text{Noun} \rightarrow & \text{cat, cats, dog, dogs, I, you, we, they, he, she, it} \\
\text{Det} \rightarrow & \text{the, a, this, these, some, many} \\
\text{Verb} \rightarrow & \text{bark, barks, barked, am, is, are, was, were,} \\
 & \text{rely, relies, see, sees, saw} \\
\text{Prep} \rightarrow & \text{on, in, above, before}
\end{array}
$$

Problems with CFG (with atomic node labels)

- How could that be fixed, using the CFG formalism?

$$
\begin{aligned}
\mathrm{S} &\rightarrow && \mathrm{NP\_sg\ VP\_sg} \\
\mathrm{S} &\rightarrow && \mathrm{NP\_pl\ VP\_pl} \\
\mathrm{NP\_sg} &\rightarrow && (\mathrm{Det\_sg})\ \mathrm{Noun\_sg} \\
\mathrm{NP\_pl} &\rightarrow && (\mathrm{Det\_pl})\ \mathrm{Noun\_pl} \\
\mathrm{VP\_sg} &\rightarrow && \mathrm{V\_intrans\_sg} \\
\mathrm{VP\_sg} &\rightarrow && \mathrm{V\_trans\_sg\ NP\_sg} \\
\mathrm{VP\_sg} &\rightarrow && \mathrm{V\_trans\_sg\ NP\_pl} \\
&\ldots
\end{aligned}
$$

Generalized Phrase Structure Grammar (GPSG)

- Gazdar et al 1982
- Added feature structures to CFG, but stayed CFG-equivalent
- Eventually, it became generally accepted that natural languages are in fact not context free
- GPSG generalized to HPSG (Pollard & Sag 1994)

Feature Structures

- Break ‘atomic’ symbols like ‘V_intrans_sg’ into bundles of information
- Allows for the statement of cross-cutting generalizations

	3rd singular subject	plural subject
direct object NP	*denies*	*deny*
no direct object NP	*disappears*	*disappear*

Attribute value matrices

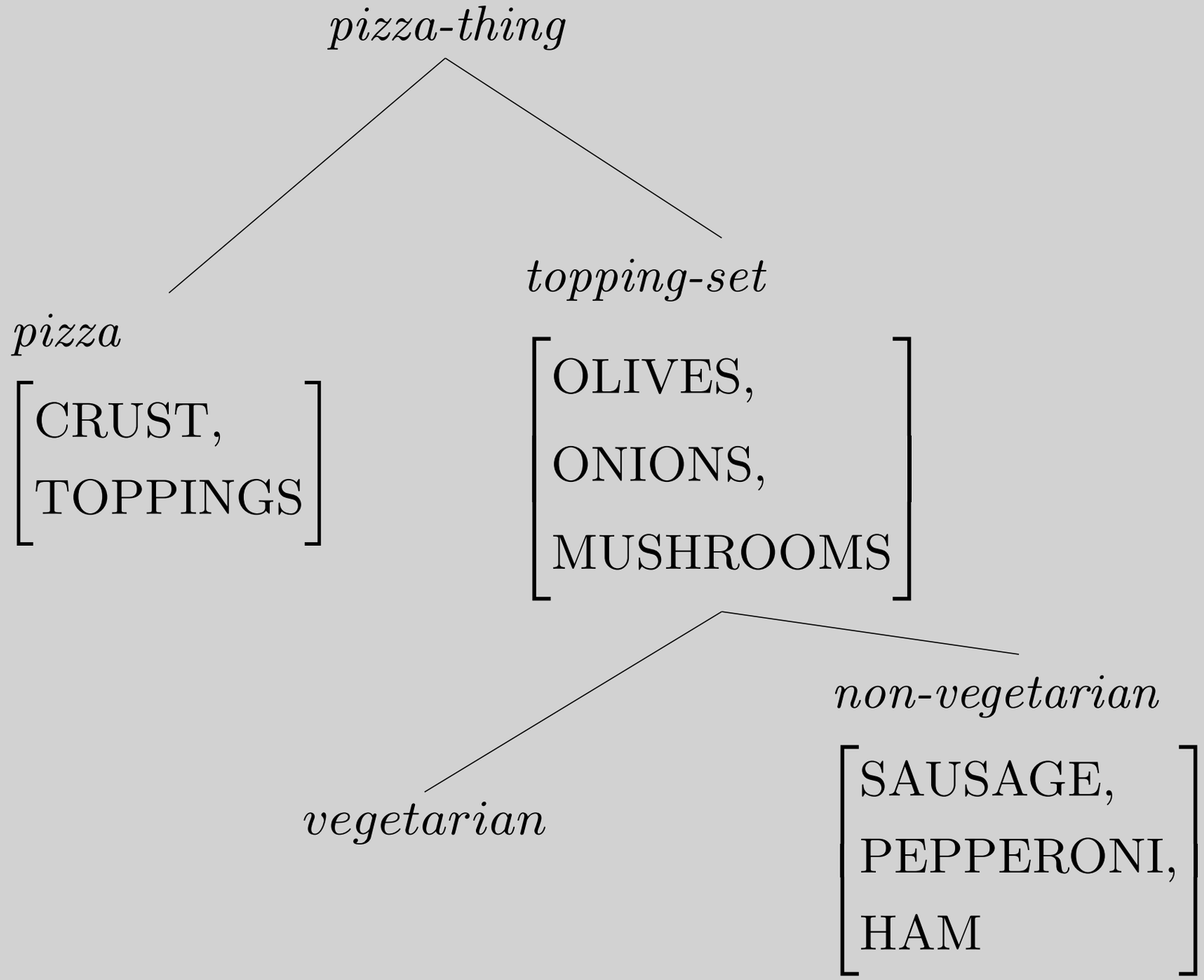
$$\begin{bmatrix} \text{FEATURE}_1 & \text{VALUE}_1 \\ \text{FEATURE}_2 & \text{VALUE}_2 \\ \multicolumn{2}{c}{\ldots} \\ \text{FEATURE}_n & \text{VALUE}_n \end{bmatrix}$$

- Values can be atomic symbols, or feature structures in their own right.

Unification

- Test whether two feature structures are compatible
- If so, find the most general feature structure that includes all information from both
- Section 15.2 shows unification of untyped feature structures
- Pizza examples (following) add in types (see 15.6)

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

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

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?

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

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

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

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

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

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

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

Combining Constraints

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

$$=$$

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

$$\left[\begin{array}{ll} \textit{pizza} & \\ \text{ONE-HALF} & \boxed{1}\left[\begin{array}{ll}\text{ONIONS} & +\\ \text{OLIVES} & -\end{array}\right] \\ \text{OTHER-HALF} & \boxed{1} \end{array}\right] \;\&\; \left[\begin{array}{ll} \textit{pizza} & \\ \text{OTHER-HALF} & \left[\begin{array}{ll}\text{MUSHROOMS} & -\\ \text{OLIVES} & -\end{array}\right] \end{array}\right]$$

Combining Constraints

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

$$=$$

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

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

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

Combining Constraints

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

$$= \phi$$

Fixing the unwieldy grammar: Agreement

- Awkward CFG analyses

$$
\begin{aligned}
\text{S} &\rightarrow && \text{NP\_sg VP\_sg} \\
\text{S} &\rightarrow && \text{NP\_pl VP\_pl} \\
\text{NP\_sg} &\rightarrow && \text{(Det\_sg) Noun\_sg} \\
\text{NP\_pl} &\rightarrow && \text{(Det\_pl) Noun\_pl} \\
\text{VP\_sg} &\rightarrow && \text{V\_intrans\_sg} \\
\text{VP\_sg} &\rightarrow && \text{V\_trans\_sg NP\_sg} \\
\text{VP\_sg} &\rightarrow && \text{V\_trans\_sg NP\_pl} \\
&\ldots
\end{aligned}
$$

Fixing the unwieldy grammar: Agreement

- Better, with unification:

$$
\begin{aligned}
\mathrm{S} \rightarrow & \quad \mathrm{NP}[\mathrm{AGR}\ \boxed{1}]\ \mathrm{VP}[\mathrm{AGR}\ \boxed{1}] \\
\mathrm{NP}[\mathrm{AGR}\ \boxed{1}] \rightarrow & \quad (\mathrm{Det}[\mathrm{AGR}\ \boxed{1}])\ \mathrm{Noun}[\mathrm{AGR}\ \boxed{1}] \\
\mathrm{VP}[\mathrm{AGR}\ \boxed{1}] \rightarrow & \quad \mathrm{V\_intrans}[\mathrm{AGR}\ \boxed{1}] \\
\mathrm{VP}[\mathrm{AGR}\ \boxed{1}] \rightarrow & \quad \mathrm{V\_trans}[\mathrm{AGR}\ \boxed{1}]\ \mathrm{NP} \\
\mathrm{VP}[\mathrm{AGR}\ \boxed{1}] \rightarrow & \quad \mathrm{V\_pp\_trans}[\mathrm{AGR}\ \boxed{1}]\ \mathrm{PP} \\
\mathrm{VP}[\mathrm{AGR}\ \boxed{1}] \rightarrow & \quad \mathrm{V\_ditrans}[\mathrm{AGR}\ \boxed{1}]\ \mathrm{NP}\ \mathrm{NP} \\
\mathrm{VP}[\mathrm{AGR}\ \boxed{1}] \rightarrow & \quad \mathrm{V\_pp\_ditrans}[\mathrm{AGR}\ \boxed{1}]\ \mathrm{NP}\ \mathrm{PP} \\
\ldots &
\end{aligned}
$$

Fixing the unwieldy grammar: Subcategorization

$$\text{S} \rightarrow \text{NP}[\text{AGR } \boxed{1}]\ \text{VP}[\text{AGR } \boxed{1}]$$
$$\text{NP}[\text{AGR } \boxed{1}] \rightarrow (\text{Det}[\text{AGR } \boxed{1}])\ \text{Noun}[\text{AGR } \boxed{1}]$$
$$\text{VP}[\text{AGR } \boxed{1}] \rightarrow \text{V}[\text{AGR } \boxed{1}, \text{SUBCAT } \boxed{A}]\ \boxed{A}$$
$$\text{V}[\text{AGR sg}, \text{SUBCAT } \langle\ \rangle] \rightarrow \text{sleeps}$$
$$\text{V}[\text{AGR pl}, \text{SUBCAT } \langle\ \rangle] \rightarrow \text{sleep}$$
$$\text{V}[\text{SUBCAT } \langle\ \rangle] \rightarrow \text{slept}$$
$$\text{V}[\text{AGR sg}, \text{SUBCAT } \langle\ NP\rangle] \rightarrow \text{sees}$$
$$\text{V}[\text{AGR pl}, \text{SUBCAT } \langle\ NP\rangle] \rightarrow \text{see}$$
$$\text{V}[\text{SUBCAT } \langle\ NP\rangle] \rightarrow \text{saw}$$
$$\ldots$$

Examples

- *wh*-questions:

  What did you find?

  Tell me who you talked to

- relative clauses:

  the item that I found

  the guy who(m) I talked to

- topicalization:

  The manual, I can't find

  Chris, you should talk to.

- *easy*-adjectives:

  My house is easy to find.
  Pat is hard to talk to.

What these have in common

- There is a ‘gap’: nothing following *find* and *to*, even though both normally require objects.
- Something that fills the role of the element missing from the gap occurs at the beginning of the clause.
- We use topicalization and *easy*-adjectives to illustrate:

  The manual, I can’t find _____

  Chris is easy to talk to _____

Gaps and their fillers can be far apart:

- *<u>The solution to this problem</u>, Pat said that someone claimed you thought I would never find____.*

- *<u>Chris</u> is easy to consider it impossible for anyone but a genius to try to talk to____.*

☞ That's why we call them "long distance dependencies"

Fillers often have syntactic properties associated with their gaps

Him, I haven't met___.

**He, I haven't met___.*

The scissors, Pat told us ____ were missing.

**The scissors, Pat told us ____ was missing.*

On Pat, you can rely___.

**To Pat, you can rely___.*

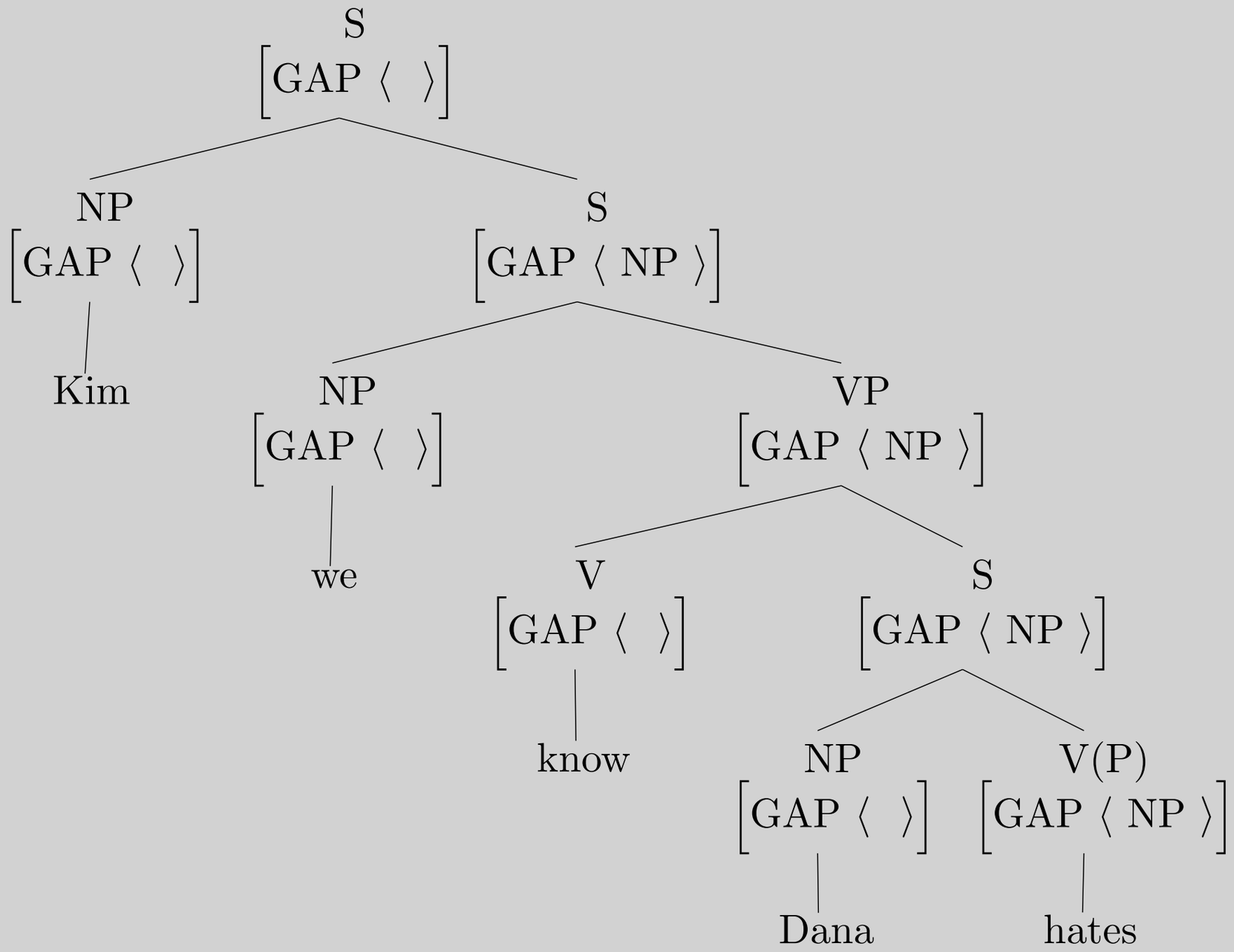
Very Rough Sketch of Our Approach

- A feature GAP records information about a missing constituent.
- The GAP value is passed up the tree by a new principle.
- A new grammar rule expands S as a filler followed by another S whose GAP value matches the filler.
- Caveat: Making the details of this general idea work involves several complications.

A Word with a Non-Empty GAP Value

$$\left\langle \text{hand} , \left[\begin{array}{l} \textit{word} \\ \\ \text{SYN} \left[\begin{array}{ll} \text{HEAD} & \left[\text{FORM} \quad \text{fin} \right] \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle\, \boxed{1} \,\rangle \\ \text{COMPS} & \langle\, \boxed{3}\text{PP[to]} \,\rangle \end{array} \right] \\ \text{GAP} & \langle\, \boxed{2}\text{NP[acc]} \,\rangle \end{array} \right] \\ \\ \text{ARG-ST} \quad \left\langle \boxed{1}\text{NP} \left[\begin{array}{ll} \text{CASE} & \text{nom} \\ \text{AGR} & \textit{non-3sing} \end{array} \right] , \boxed{2} , \boxed{3} \right\rangle \end{array} \right] \right\rangle$$

How We Want GAP to Propagate



The Head-Filler Rule

$$[phrase] \rightarrow \boxed{1}\begin{bmatrix}\text{GAP} & \langle\ \rangle\end{bmatrix} \quad \mathbf{H}\begin{bmatrix}\text{HEAD} & \begin{bmatrix} verb \\ \text{FORM} & \text{fin}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix} \\ \text{STOP-GAP} & \langle\ \boxed{1}\ \rangle \\ \text{GAP} & \langle\ \boxed{1}\ \rangle\end{bmatrix}$$

Overview

- Problems with CFG
- Feature structures
- Unification
- Agreement
- Subcategorization
- Long-distance Dependencies
- Reading questions

Reading questions

- I know what regular and non-regular grammars are, but what makes a grammar context-free?

- There is a distinction made between right and left linear grammar, but I don't really understand why this distinction is too important; is it just for the sake of writing proofs? Shouldn't there also be consideration for terminals appearing on both the left, right and the middle?

Reading questions

- I was not really clear about the two forms of specified constraints:
  - <Bi feature path> = Atomic value
  - <Bi feature path> = <Bi feature path>
- How should I understand these two forms? Also, "the notation <Bi feature path> denotes a feature path through the feature structure associated with the Bi component of the context-free part of the rule", this sentence seems helpful for understanding the forms but it is really confusing to me, how should I understand it?

Reading questions

- I'm confused as to how the feature system saves you very much from a proliferation of grammar rules. Don't you still have to write separate grammar rules for each form of each feature?

- Does the order of the features in the attribute-value matrix matter? In the readings, some have [PERSON 3, then NUMBER sg] while others have [NUMBER sg, then PERSON 3]. There is one example on pg 495 where after unification, [PERSON 3, NUMBER sg] becomes [NUMBER sg, PERSON 3]. Do the order differ because of which is more significant or are they just random?

- I'm confused as to how auxiliaries function in these agreement patterns. It seems that auxiliaries take away some of the agreement functionality of the main verbs- in the sentence "He is running," only "is" has 3rd / sg agreement features, not the main verb "running". How would it look for full agreement functionality to be written out for a sentence with an auxiliary?

Reading questions

- I was wondering how feature structures tend to vary with the "type" of language. Would word order based languages lead to more, smaller feature structures, and polysynthetic languages lead to fewer, larger feature structures?

- How do feature structures work in analyzing transcriptions of natural speech? Often times, when we talk to one another, we'll leave out dependencies because they are included in wider context. Can feature structures account for this?

Reading questions

- Is the gap list in long distance dependencies the reason we saw the stacked NP's and such in the parse tree we saw in class on Tuesday?

- How are feature structures encoded? Do they have to be designed by hand for each word? Have there been attempts to use machine learning to deduce argument structures for verbs, for example?