October 21, 2003

Chapter 11.1–11.2

Feature structures

Summary

- Problems with CFG
- One solution: feature structure and unification
- An aside on modeling language
- Practice with unification (pizza)
- Concepts relating to feature structures
- Feature structures in the grammar
- Using types to capture linguistic generalizations

Problems with CFG

- Potentially arbitrary rules: D $\rightarrow$ S VP
- Gets clunky quickly with cross-cutting properties (we'll return to this)
- (Not quite powerful enough for natural languages)

Solution: Replace atomic node labels with feature structures.

Some grammatical theories which explicitly use feature structures and unification

- GPSG: Generalized Phrase Structure Grammar
- HPSG: Head-driven Phrase Structure Grammar

  http://hpsg.stanford.edu/
- LFG: Lexical Functional Grammar

  http://www-lfg.stanford.edu/lfg/
- Construction Grammar

  http://www.constructiongrammar.org/
- Unification Categorial Grammar

Modeling language (HPSG)

Two kinds of modeling:

- Speakers' internalized grammars
- Set of possible English sentences

Modeling language (HPSG)

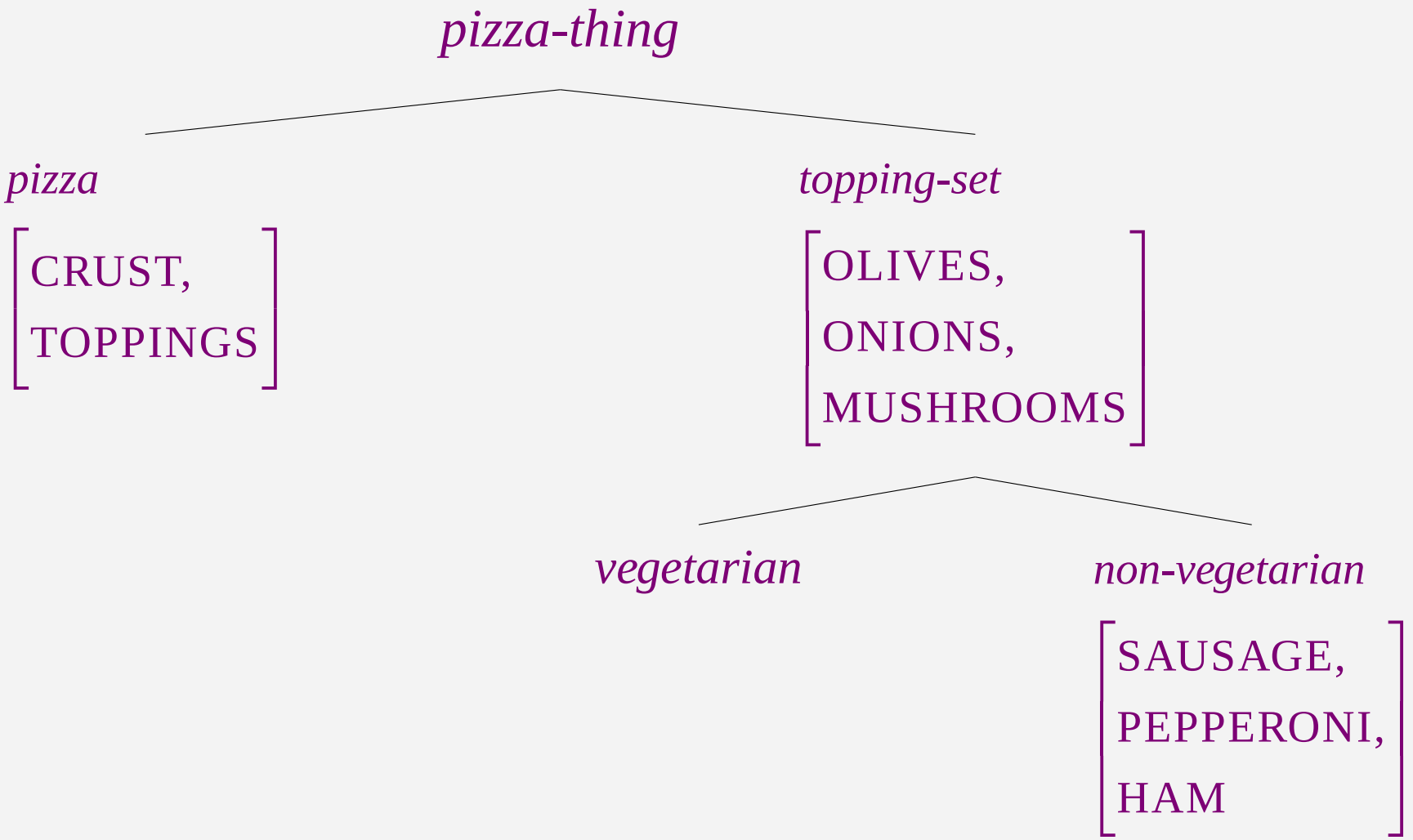
Three things involved in modeling:

- Real-world entities
- Models
- Descriptions of models

Feature Structure Descriptions

$$\begin{bmatrix} \text{FEATURE}_1 & \mathit{VALUE}_1 \\ \text{FEATURE}_2 & \mathit{VALUE}_2 \\ \cdots & \\ \text{FEATURE}_n & \mathit{VALUE}_n \end{bmatrix}$$

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	$\begin{bmatrix} \text{SAUSAGE} & \{+, -\} \\ \text{PEPPERONI} & \{+, -\} \\ \text{BBQ CHICKEN} & \{+, -\} \end{bmatrix}$	*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 IST relations).
- ... states what general properties each kind of object has (the feature and feature value declarations).

Pizza Descriptions and Pizza Models

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

How many fully resolved pizza models satisfy this description?

Pizza Descriptions and Pizza Models

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

$\{ \langle \text{CRUST} , \text{thick} \rangle , \langle \text{TOPPINGS} , \{ \langle \text{OLIVES} , + \rangle , \langle \text{ONIONS} , + \rangle , \langle \text{MUSHROOMS} , - \rangle \} \rangle \}$

$\{ \langle \text{CRUST} , \text{thick} \rangle , \langle \text{TOPPINGS} , \{ \langle \text{OLIVES} , + \rangle , \langle \text{ONIONS} , + \rangle , \langle \text{MUSHROOMS} , + \rangle \} \rangle \}$

Pizza Descriptions and Pizza Models

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

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

'type'/'token' distinction – applies to sentences as well

Unification

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

Unification

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

Unification

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

Unification

ϕ

Unification

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

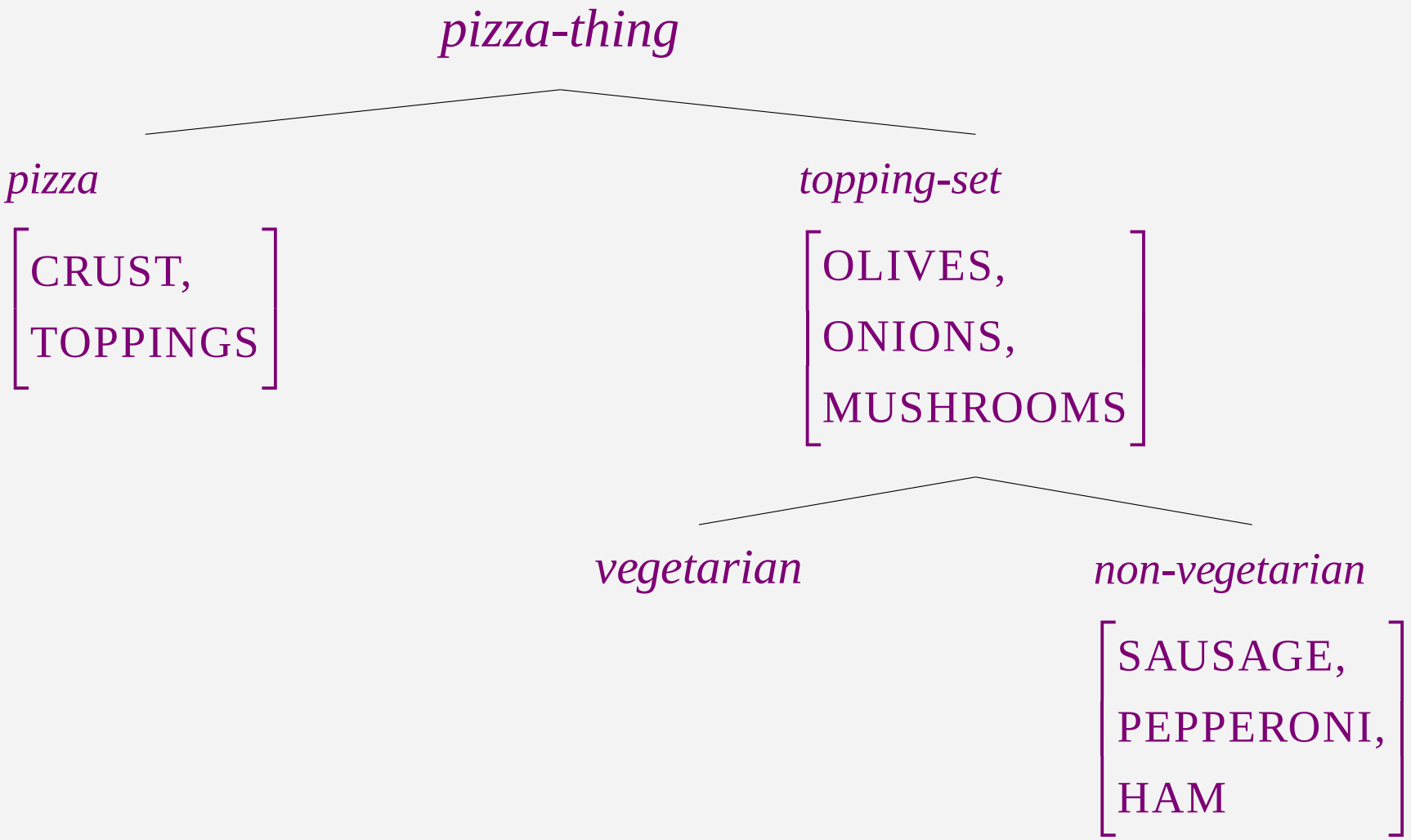
Unification

Unification

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

Unification

A Pizza Type Hierarchy



A New Theory of Pizzas

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

Unification

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

Unification

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

Identity Constraints (Tags)

$$
\begin{bmatrix}
\textit{pizza} & \\
\text{CRUST} & \textit{thin} \\
\text{ONE-HALF} & \begin{bmatrix} \text{OLIVES} & \boxed{1} \\ \text{ONIONS} & \boxed{2} \end{bmatrix} \\
\text{OTHER-HALF} & \begin{bmatrix} \text{OLIVES} & \boxed{1} \\ \text{ONIONS} & \boxed{2} \end{bmatrix}
\end{bmatrix}
$$

Unification

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

Unification

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

Unification

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

Unification

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

Unification

Concepts relating to feature structures

- Underspecification
- Subsumption
  - Defines a partial-order
  - Mutual non-subsumption does not entail incompatibility
  - Mutual subsumption does entail equality
- Monotonicity
- Order independence

Feature structures as DAGs

- Features label arcs
- Nodes are associated with sub-DAGs or atomic values
- With types, each node is labeled with a type
- Identity constraints represented by reentrancy

Feature structures in the grammar

- Associate complex feature structures with both lexical items and instances of grammatical categories.
- Guide the composition of feature structures for larger grammatical constituents based on the feature structures of their component parts.
- Enforce compatibility constraints between specified parts of grammatical constructions.

Example 1: Subject-verb agreement

- What does it mean for the subject and the verb to agree?
- How would we handle agreement in plain CFG?
- What kind of information would it be useful to encode in features?
- Two ways to make the rules use the features:
  - Multiple rules, with feature specified
  - Identity constraints

Example 2: Subcategorization

- What is subcategorization (e.g., verb subcategorization)?
- How would we handle agreement in plain CFG?
- What kind of information would it be useful to encode in features?
- Two ways to make the rules use the features:
  - Multiple rules, with feature specified
  - Identity constraints

Example 3: Subcategorization and agreement

- How would we handle both agreement and subcategorization together in plain CFG?
- Are our new rules for each compatible already, or do they need modification?

Using types to capture linguistic generalizations

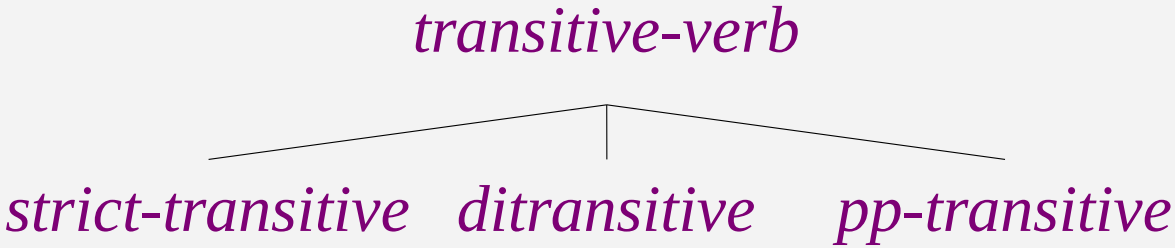
- The lexicon is both the repository of idiosyncrasy and full of redundancy.
- Rather than state

  [VAL tr] or [SUBCAT ⟨ NP ⟩]

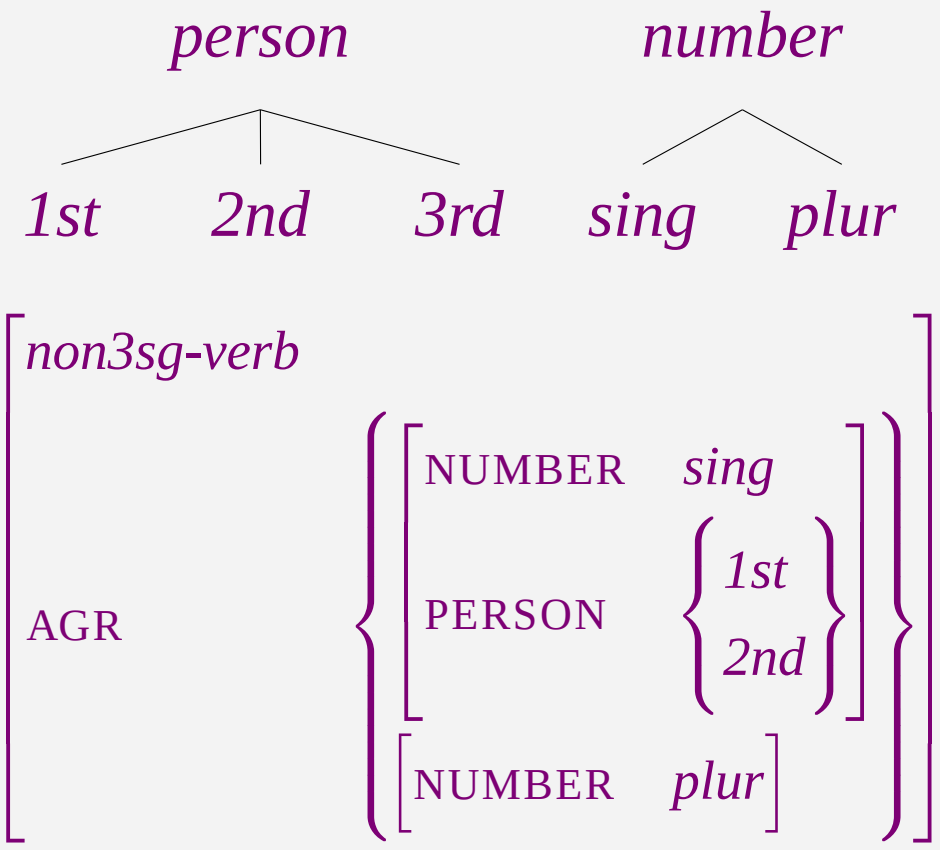
  on every single transitive verb,
- Create a type *transitive-verb*, subject to that constraint, and let individual verbs be instances of that type.

Using types to capture linguistic generalizations

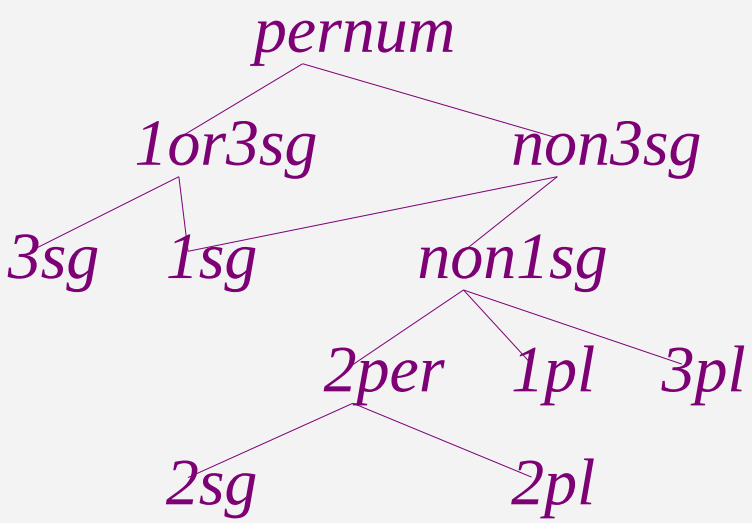
- Allow for intermediate generalizations as well:



Using types to capture linguistic generalizations



Using types to capture linguistic generalizations



$$\begin{bmatrix} \textit{non3sg-verb} & \\ \text{AGR} & \begin{bmatrix} \text{PERNUM} & \textit{non3sg} \end{bmatrix} \end{bmatrix}$$

Summary

- Feature structures can be used to express finer-grained details within (grammatical) categories.
- Feature structures can be combined with unification.
- Feature structures allow grammar writers to capture more generalizations.
- Types allow grammar writers to capture even more generalizations.

Next time

- Implementing unification
- Integrating unification into the parser
- More linguistic examples