Ling 566
Oct 28, 2010
Lexical Types
Overview

• Motivation for lexical hierarchy
• Default inheritance
• Tour of the lexeme hierarchy
• The Case Constraint
• pos vs. lexeme
Motivation

• We've streamlined our grammar rules...
  • ...by stating some constraints as general principles
  • ...and locating lots of information in the lexicon.
  • Our lexical entries currently stipulate a lot of information that is common across many entries and should be stated only once.

• Examples?

• Ideally, particular lexical entries need only give phonological form, the semantic contribution, and any constraints truly idiosyncratic to the lexical entry.
Lexemes and Words

- **Lexeme**: An abstract proto-word which gives rise to genuine words. We refer to lexemes by their ‘dictionary form’, e.g. ‘the lexeme run’ or ‘the lexeme dog’.

- **Word**: A particular pairing of form and meaning. *Running* and *ran* are different words.
Lexical Types & Lexical Rules

- Lexemes capture the similarities among run, runs, running, and run.

- The lexical type hierarchy captures the similarities among run, sleep, and laugh, among those and other verbs like devour and hand, and among those and other words like book.

Q: What do devour and book have in common?
A: The SHAC

- Lexical rules capture the similarities among runs, sleeps, devours, hands,...
Default Inheritance

Q: Why do we have default inheritance?

A: Generalizations with exceptions are common:

- Most nouns in English aren't marked for CASE, but pronouns are.
- Most verbs in English only distinguish two agreement categories ($3$sing and non-$3$sing), but be distinguishes more.
- Most prepositions in English are transitive, but here and there are intransitive.
- Most nominal words in English are 3rd person, but some (all of them pronouns) are 1st or 2nd person.
- Most proper nouns in English are singular, but some (mountain range names, sports team names) are plural.
Default Inheritance, Technicalities

If a type says ARG-ST / < NP >, and one of its subtypes says ARG-ST < >, then the ARG-ST value of instances of the subtype is < >.

If a type says ARG-ST < NP >, and one of its subtypes says ARG-ST < >, then this subtype can have no instances, since they would have to satisfy contradictory constraints.
If a type says MOD / < S >, and one of its subtypes says MOD <[SPR < NP> ]>, then the ARG-ST value of instances of the subtype is what?

\[
\begin{align*}
\text{MOD} & \left\langle \begin{bmatrix}
\text{HEAD} & \text{verb} \\
\text{SPR} & \langle \text{NP} \rangle \\
\text{COMPS} & \langle \rangle
\end{bmatrix}\right\rangle
\end{align*}
\]

That is, default constraints are ‘pushed down’
Question on Default Inheritance

Q: Can a grammar rule override a default constraint on a word?

A: No. Defaults are all ‘cached out’ in the lexicon.

• Words as used to build sentences have only inviolable constraints.
Our Lexeme Hierarchy

```plaintext
synsem [SYN, SEM]

lexeme [ARG-ST]

expression [ARG-ST]

word phrase

const-lxm

infl-lxm

verb-lxm cn-lxm

siv-lxm piv-lxm tv-lxm cntn-lxm massn-lxm

stv-lxm dtv-lxm ptv-lxm
```
Functions of Types

• Stating what features are appropriate for what categories

• Stating generalizations
  • Constraints that apply to (almost) all instances
  • Generalizations about selection -- where instances of that type can appear
Every \textit{synsem} has the features \textit{SYN} and \textit{SEM}
No ARG-ST on phrase
A Constraint on $infl-lxm$: the SHAC

- $synsem$ [SYN, SEM]
- $lexeme$ [ARG-ST]
- $expression$
  - $word$ [ARG-ST]
  - $phrase$
- $infl-lxm$
- $const-lxm$
- $verb-lxm$
  - $siv-lxm$
  - $piv-lxm$
  - $tv-lxm$
  - $cntn-lxm$
  - $massn-lxm$
- $cn-lxm$
- $adj-lxm$
- $conj-lxm$
- $det-lxm$
- $predp-lxm$
- $argmkp-lxm$
- $pn-lxm$
- $pron-lxm$
A Constraint on \textit{infl-lxm}: the SHAC

\[
\text{infl-lxm} : \left[ \begin{array}{c}
\text{SYN} \\
\text{VAL} \\
\text{HEAD}
\end{array} \right] \left[ \begin{array}{c}
\text{SPR} \left[ \begin{array}{c}
\text{AGR} \\
1
\end{array} \right]
\end{array} \right]
\]
Constraints on \textit{cn-lxm}
Constraints on \( cn-lxm \)

\[
\begin{array}{c}
\text{SYN} \\
\text{VAL} \\
\text{SEM} \\
\text{ARG-ST}
\end{array}
\]

\[
\begin{array}{c}
\text{HEAD} \\
\text{AGR} \\
\text{SPR} \\
\text{MODE} \\
\text{INDEX} \\
\langle X \rangle \oplus / \langle \rangle \\
\end{array}
\]

\[
\begin{array}{c}
\text{noun} \\
\text{PER 3rd} \\
\text{det} \\
\text{i} \\
\text{i}
\end{array}
\]
Formally Distinguishing Count vs. Mass Nouns
Formally Distinguishing Count vs. Mass Nouns

\[
\text{cntn-}l\text{xm} : \left[ \text{SYN} \left[ \text{VAL} \left[ \text{SPR} \left( [\text{COUNT} +] \right) \right] \right] \right]
\]

\[
\text{massn-}l\text{xm} : \left[ \text{SYN} \left[ \text{VAL} \left[ \text{SPR} \left( [\text{COUNT} -] \right) \right] \right] \right]
\]
Constraints on \textit{verb-lxm}

\begin{itemize}
  \item \textit{synsem} \hspace{1cm} \textit{lexeme} \hspace{1cm} \textit{expression}
  \item \textit{infl-lxm} \hspace{1cm} \textit{const-lxm} \hspace{1cm} \textit{word} \hspace{1cm} \textit{phrase}
  \item \textit{verb-lxm} \hspace{1cm} \textit{cn-lxm} \hspace{1cm} \textit{pn-lxm} \hspace{1cm} \textit{pron-lxm}
  \item \textit{siv-lxm} \hspace{1cm} \textit{piv-lxm} \hspace{1cm} \textit{tv-lxm} \hspace{1cm} \textit{cntn-lxm} \hspace{1cm} \textit{massn-lxm}
  \item \textit{stv-lxm} \hspace{1cm} \textit{dtv-lxm} \hspace{1cm} \textit{ptv-lxm}
\end{itemize}
Constraints on \textit{verb-lxm}

\begin{align*}
\text{\textit{verb-lxm}:} & \quad \begin{bmatrix}
\text{SYN} & \begin{bmatrix}\text{HEAD} & \text{verb}\end{bmatrix} \\
\text{SEM} & \begin{bmatrix}\text{MODE} & \text{prop}\end{bmatrix} \\
\text{ARG-ST} & / \langle \text{NP}, \ldots \rangle
\end{bmatrix}
\end{align*}
Subtypes of *verb-lxm*

- **verb-lxm**: \[ARG-ST / < NP, ... >\]
- **siv-lxm**: \[ARG-ST / < NP >\]
- **piv-lxm**: \[ARG-ST / < NP, PP >\]
- **tv-lxm**: \[ARG-ST / < NP, NP, ... >\]
- **stv-lxm**: \[ARG-ST / < NP, NP, >\]
- **dtv-lxm**: \[ARG-ST / < NP, NP, NP >\]
- **ptv-lxm**: \[ARG-ST / < NP, NP, PP >\]
Proper Nouns and Pronouns

synsem
[SYN, SEM]

lexeme
[ARG-ST]

expression

word

phrase

const-lxm

pn-lxm

pron-lxm

adj-lxm
	con-lxm

det-lxm

predp-lxm
	argmkp-lxm

ing-lxm

verb-lxm
	cn-lxm

siv-lxm
	
piv-lxm
	
tv-lxm
	
cntn-lxm
	
massn-lxm

stv-lxm

dtv-lxm

dtv-lxm

© 2003 CSLI Publications
Proper Nouns and Pronouns

$pn-lxm:\[
\begin{array}{c}
\text{SYN} \quad \text{HEAD} \quad \text{noun} \\
\text{SEM} \quad \text{MODE} \quad \text{ref} \\
\text{ARG-ST} \quad / \langle \rangle \\
\end{array}
\]

$pron-lxm:\[
\begin{array}{c}
\text{SYN} \quad \text{HEAD} \quad \text{noun} \\
\text{SEM} \quad \text{MODE} \quad / \text{ref} \\
\text{ARG-ST} \quad \langle \rangle \\
\end{array}
\]
The Case Constraint

An outranked NP is [CASE acc].

- object of verb ✓
- second object of verb ✓
- object of argument-marking preposition ✓
- object of predicational preposition (✓)
The Case Constraint, continued

An outranked NP is [CASE acc].

- Subjects of verbs
  - Should we add a clause to cover nominative subjects?
    - No.

  We expect them to leave.  (Chapter 12)

  - Lexical rules for finite verbs will handle nominative subjects.

- Any other instances of case marking in English?

- Does it apply to case systems in other languages?
  No: The Case Constraint is an English-specific constraint.
Apparent redundancy

• Why do we need both the *pos* subhierarchy and lexeme types?

• *pos*:
  • Applies to words and phrases; models relationship between then
  • Constrains which features are appropriate (no AUX on *noun*)

• *lexeme*:
  • Generalizations about combinations of constraints
Lexical Types & Lexical Rules

• Lexemes capture the similarities among run, runs, running, and run.

• The lexical type hierarchy captures the similarities among run, sleep, and laugh, among those and other verbs like devour and hand, and among those and other words like book.

• Lexical rules capture the similarities among runs, sleeps, devours, hands,...
Overview

- Motivation for lexical hierarchy
- Default inheritance
- Tour of the lexeme hierarchy
- The Case Constraint
- pos vs. lexeme