Ling 566
Oct 25, 2007
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 (3sing and non-3sing), 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.
Default Inheritance, More Technicalities

• 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?

• 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

\[
\text{synsem} \\
\text{[SYN, SEM]}
\]

\[
\text{lexeme} \\
\text{[ARG-ST]}
\]

\[
\text{expression} \\
\text{word} \\
\text{phrase}
\]

\[
\text{infl-lxm}
\]

\[
\text{const-lxm}
\]

\[
\text{pn-lxm} \\
\text{pron-lxm}
\]

\[
\text{adj-lxm} \\
\text{conj-lxm} \\
\text{det-lxm} \\
\text{predp-lxm} \\
\text{argmkp-lxm}
\]

\[
\text{verb-lxm}
\]

\[
\text{cn-lxm}
\]

\[
\text{siv-lxm} \\
\text{piv-lxm} \\
\text{tv-lxm} \\
\text{cnln-lxm} \\
\text{massn-lxm}
\]

\[
\text{stv-lxm} \\
\text{dte-lxm} \\
\text{ptv-lxm}
\]
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 *synsem* has the features SYN and SEM
No ARG-ST on phrase

```
synsem
[SYN, SEM]

lexeme
[ARG-ST]

expression

word
phrase

infl-lxm

const-lxm

verb-lxm

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

adj-lxm conj-lxm det-lxm predp-lxm argmkp-lxm

pn-lxm pron-lxm

cn-lxm

stv-lxm dtv-lxm ptv-lxm```

© 2003 CSLI Publications
A Constraint on \textit{infl-lxm}: the SHAC
A Constraint on \( infl-lxm \): the SHAC

\[
infl-lxm : \begin{bmatrix}
\text{SYN} & \begin{bmatrix} \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle [\text{AGR} & [1] \rangle \rangle \\ \text{HEAD} & [\text{AGR} & [1] \rangle \rangle \\ & \end{bmatrix} \\ & \end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\]
Constraints on \( cn-lxm \)
Constraints on $cn\text{-}lxm$

$$cn\text{-}lxm : \begin{bmatrix}
\text{SYN} & \begin{bmatrix}
\text{HEAD} & \begin{bmatrix}
\text{noun} & \text{AGR} & \text{[PER 3rd]} \\
\text{SPR} & \langle \begin{bmatrix}
\text{HEAD} & \text{det} \\
\text{INDEX} & i 
\end{bmatrix} \rangle \\
\text{MODE} & / \text{ref} \\
\text{INDEX} & i
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\text{VAL} & \begin{bmatrix}
\langle X \rangle \oplus / \langle \rangle 
\end{bmatrix}
\text{SEM} & \begin{bmatrix}
\langle X \rangle \oplus / \langle \rangle 
\end{bmatrix}
\text{ARG-ST} & \begin{bmatrix}
\langle X \rangle \oplus / \langle \rangle 
\end{bmatrix}
\end{bmatrix}$$
Formally Distinguishing Count vs. Mass Nouns
Formally Distinguishing Count vs. Mass Nouns

cntn-lxm : \[ SYN \left[ VAL \left[ SPR \langle [COUNT +] \rangle \right] \right] \]

massn-lxm : \[ SYN \left[ VAL \left[ SPR \langle [COUNT -] \rangle \right] \right] \]
Constraints on *verb-lxm*
Constraints on \textit{verb-lxm}

\[
\begin{array}{c}
\text{SYN} & [\text{HEAD} \quad \text{verb}] \\
\text{SEM} & [\text{MODE} \quad \text{prop}] \\
\text{ARG-ST} & / \langle \text{NP, ...} \rangle \\
\end{array}
\]
Subtypes of $verb$-$lxm$

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

```
  synsem
      [SYN, SEM]

  lexeme
      [ARG-ST]

  expression

  infl-lxm

  const-lxm

  pn-lxm  pron-lxm

  adj-lxm  conj-lxm  det-lxm  predp-lxm  argmkp-lxm

  verb-lxm

  cn-lxm

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

  stv-lxm  dtv-lxm  ptv-lxm
```
Proper Nouns and Pronouns

\[
\begin{align*}
\text{pn-}l_{xm} & : \\
\text{SYN} & : \left[ \begin{array}{c}
\text{HEAD} \\
noun
\end{array} \right] \\
\text{AGR} & : \left[ \begin{array}{c}
\text{PER} \\
\text{3rd}
\end{array} \right] \\
\text{NUM} & : \left[ \begin{array}{c}
\text{sg}
\end{array} \right]
\end{align*}
\]

\[
\begin{align*}
\text{SEM} & : \left[ \begin{array}{c}
\text{MODE} \\
\text{ref}
\end{array} \right] \\
\text{ARG-ST} & : \langle \rangle
\end{align*}
\]

\[
\begin{align*}
\text{pron-}l_{xm} & : \\
\text{SYN} & : \left[ \begin{array}{c}
\text{HEAD} \\
noun
\end{array} \right]
\end{align*}
\]

\[
\begin{align*}
\text{SEM} & : \left[ \begin{array}{c}
\text{MODE} \\
\text{/ ref}
\end{array} \right] \\
\text{ARG-ST} & : \langle \rangle
\end{align*}
\]
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