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 (\textit{3sing} and \textit{non-3sing}), but \textit{be} distinguishes more.
• Most prepositions in English are transitive, but \textit{here} and \textit{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 $\langle < S > \rangle$, and one of its subtypes says MOD $\langle [\text{SPR} < \text{NP} > ] \rangle$, 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

synsem
[SYN, SEM]

lexeme
[ARG-ST]

expression

word
phrase

const-lxm

verb-lxm

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

infl-lxm

pn-lxm pron-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 *synsem* has the features SYN and SEM
No ARG-ST on phrase
A Constraint on *infl-lxm*: the SHAC
A Constraint on \( \text{infl-lxm} \): the SHAC

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

\[
\begin{align*}
\text{SYN} : \quad & \left[\begin{array}{c}
\text{HEAD} \\
\text{AGR} [\text{PER 3rd}] \\
\text{SPR} \langle \left[ \begin{array}{c}
\text{HEAD} \\
\text{INDEX} \\
\text{det} \\
i 
\end{array} \right] \rangle
\end{array} \right] \\
\text{VAL} : \quad & \left[\begin{array}{c}
\text{noun} \\
\text{SPR} \langle \left[ \begin{array}{c}
\text{HEAD} \\
\text{INDEX} \\
i 
\end{array} \right] \rangle
\end{array} \right] \\
\text{SEM} : \quad & \left[\begin{array}{c}
\text{MODE} / \text{ref} \\
\text{INDEX} \\
i 
\end{array} \right] \\
\text{ARG-ST} : \quad & \langle X \rangle \oplus / \langle \rangle 
\end{align*}
\]
Formally Distinguishing Count vs. Mass Nouns
Formally Distinguishing Count vs. Mass Nouns

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

```
  synsem
  [SYN, SEM]
     
  lexeme
  [ARG-ST]
     
  expression
     
  word
  [ARG-ST]
     
  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
```
Constraints on \( verb-lxm \)

\[
verb-lxm: \begin{bmatrix}
\text{SYN} & \begin{bmatrix} \text{HEAD} & \text{verb} \end{bmatrix} \\
\text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \end{bmatrix} \\
\text{ARG-ST} & \langle \langle \text{NP}, \ldots \rangle \rangle
\end{bmatrix}
\]
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
Proper Nouns and Pronouns

\[ pn-lxm: \begin{bmatrix}
SYN & [\text{HEAD} & [\text{noun}]] \\
SEM & [\text{MODE} & \text{ref}] \\
\text{ARG-ST} & / \langle \rangle
\end{bmatrix} \]

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

An outranked NP is \([\text{CASE acc}]\).

- 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