Lexical Types
Overview

• Motivation for lexical hierarchy
• Default inheritance
• Tour of the lexeme hierarchy
• The Case Constraint
• pos vs. lexeme
• Guest appearance by Will
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,
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 \( \text{ARG-ST} / < \text{NP} > \), and one of its subtypes says \( \text{ARG-ST} < > \), then the ARG-ST value of instances of the subtype is \( < > \).

If a type says \( \text{ARG-ST} < \text{NP} > \), and one of its subtypes says \( \text{ARG-ST} < > \), 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

\[
\begin{align*}
\text{MOD} & \begin{cases} \\
\text{HEAD} & \text{/ verb} \\
\text{SPR} & \langle \text{NP} \rangle \\
\text{COMPS} & \langle \rangle \\
\end{cases}
\end{align*}
\]

• 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

**lexeme**
[ARG-ST]

**expression**

**word**
[ARG-ST]

**phrase**

**synsem**
[SYN, SEM]

**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**

**dtr-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

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

$cn$-$lxm$:

- **SYN**
  - HEAD
  - VAL
  - SPR

- **SEM**
  - MODE
  - INDEX

- **ARG-ST**
  - $\langle X \rangle \oplus / \langle \rangle$
Formally Distinguishing Count vs. Mass Nouns

![Diagram of linguistic structure with nodes and arrows indicating relationships between lexical and syntactic categories.]
Formally Distinguishing Count vs. Mass Nouns

cntn-lxm : \[\text{SYN} \ [\text{VAL} \ [\text{SPR} \ (\text{COUNT } + \text{])}]\]

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

\[
\begin{array}{c}
verb-lxm: \\
\begin{bmatrix}
SYN & [\text{HEAD} \ \text{verb}] \\
SEM & [\text{MODE} \ \text{prop}] \\
\text{ARG-ST} & \langle \text{NP}, \ldots \rangle
\end{bmatrix}
\end{array}
\]
Subtypes of *verb-lxm*

- **verb-lxm**:  \([\text{ARG-ST} / \langle \text{NP}, ... \rangle]\)
- **siv-lxm**:  \([\text{ARG-ST} / \langle \text{NP} \rangle]\)
- **piv-lxm**:  \([\text{ARG-ST} / \langle \text{NP}, \text{PP} \rangle]\)
- **tv-lxm**:  \([\text{ARG-ST} / \langle \text{NP}, \text{NP}, ... \rangle]\)
  - **stv-lxm**:  \([\text{ARG-ST} / \langle \text{NP}, \text{NP}, \rangle]\)
  - **dtv-lxm**:  \([\text{ARG-ST} / \langle \text{NP}, \text{NP}, \text{NP} \rangle]\)
  - **ptv-lxm**:  \([\text{ARG-ST} / \langle \text{NP}, \text{NP}, \text{PP} \rangle]\)
Proper Nouns and Pronouns

`synsem` [SYN, SEM]

`lexeme` [ARG-ST]

`infl-lxm` `const-lxm`

`verb-lxm` `cn-lxm`

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

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

```
pron-lxm:
SYN       HEAD     noun
SEM       [MODE / ref]
ARG-ST    ⟨ ⟩
```

```
prn-lxm:
SYN       HEAD     noun
SEM       [MODE / ref]
ARG-ST    ⟨ ⟩
```

```

pn-lxm:
SYN       [HEAD noun]
SEM       [MODE / ref]
ARG-ST    ⟨ ⟩
```

```
```
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?

- 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
Overview

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