Chapter 8:
The Lexicon
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.

• 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* (which shares with verbs the property of agreeing with its specifier).

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

• Generalizations with exceptions are common:
  • Most nouns in English aren’t visibly marked for CASE, but pronouns are.
  • Most verbs in English only distinguish two agreement categories (3\text{sing} and non-3\text{sing}), 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 is a formal mechanism for expressing such soft generalizations.
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 instances of the subtype are:

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

• That is, default constraints are ‘pushed down’
Defaults are ‘Cashed Out’ in the Lexicon

• A grammar rule cannot override a default constraint on a word

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

synsem
[SYN, SEM]

lexeme
[ARG-ST]

expression

word
phrase

infl-lxm

const-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

© 2003 CSLI Publications
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

```
synsem
  [SYN, SEM]

  lexeme
  [ARG-ST]

  const-lxm

  expression

    word
    [ARG-ST]

    phrase

    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
```
A Constraint on \( \text{infl-lxm} \): the SHAC

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

\[
\begin{align*}
\text{SYN} & : \\
\text{VAL} & : \\
\text{SEM} & : \\
\text{ARG-ST} & :
\end{align*}
\]
Formally Distinguishing Count vs. Mass Nouns

```
synsem
[SYN, SEM]

lexeme
[ARG-ST]

expression

word
phrase

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
```

© 2003 CSLI Publications
Formally Distinguishing Count vs. Mass Nouns

\[\text{cntn-lxm} : \left[ \begin{array}{ccc}
\text{SYN} & \text{VAL} & \text{SPR} \\
\end{array} \right] \left[ \begin{array}{c} \left< \text{[COUNT +]} \right> \end{array} \right]\]

\[\text{massn-lxm} : \left[ \begin{array}{ccc}
\text{SYN} & \text{VAL} & \text{SPR} \\
\end{array} \right] \left[ \begin{array}{c} \left< \text{[COUNT −]} \right> \end{array} \right]\]
Constraints on *verb-lxm*

```
              synsem
                [SYN, SEM]
                       |
              lexeme
                  [ARG-ST]
                         |
       expression
                         |
          word
          [ARG-ST]
             |
       phrase
       ---
     p\n-lxm  pron-lxm
           |
       adj-lxm  conj-lxm  det-lxm  predp-lxm  argmkp-lxm
           |
verb-lxm
           |
infl-lxm
        ---
      siv-lxm  piv-lxm  tv-lxm  cntn-lxm  massn-lxm
      |
      cn-lxm
```

stv-lxm  dtv-lxm  ptv-lxm
Constraints on *verb-lxm*

\[
\begin{align*}
\text{verb-lxm:} & \quad \left[ \begin{array}{c}
\text{SYN} \\
\text{SEM} \\
\text{ARG-ST}
\end{array} \right] \\
& \quad \left[ \begin{array}{c}
\text{HEAD} \quad \text{verb} \\
\text{MODE} \quad \text{prop} \\
/ \langle \text{NP}, \ldots \rangle
\end{array} \right]
\end{align*}
\]
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
Proper Nouns and Pronouns

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

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

• Case in English is quite rudimentary: only two cases, which only show up in personal pronouns.

• Accusative seems to be the default: nominative shows up only on the subject of a finite clause (and, in some dialects, coordinated pronouns).

• A stab at stating this in our theory is the Case Constraint: An outranked NP is [CASE  acc]

• Lexical rules for finite verbs will handle nominative subjects.
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 Feature FORM

• Different inflected forms of verbs are syntactically distinguished by the feature FORM, as assigned by lexical rules.

• FORM will also be useful in our analyses of coordination, PP selection, and non-referential NPs.

• In the literature, distinct features are employed for these different uses, but for our purposes, we can lump them together.
Coordination and FORM

Pat watched Chris run.  Pat watched Chris jump.
Pat watched Chris running.  Pat watched Chris jumping.

Pat watched Chris run and jump.
Pat watched Chris running and jumping.
*Pat watched Chris run and jumping.
*Pat watched Chris running and jump.

Coordination Rule (Chapter 8 Version):

\[
\begin{bmatrix}
\text{FORM} & 1 \\
\text{VAL} & 0 \\
\text{IND} & s_0
\end{bmatrix} \rightarrow
\begin{bmatrix}
\text{FORM} & 1 \\
\text{VAL} & 0 \\
\text{IND} & s_1
\end{bmatrix} \ldots
\begin{bmatrix}
\text{FORM} & 1 \\
\text{VAL} & 0 \\
\text{IND} & s_{n-1}
\end{bmatrix}
\begin{bmatrix}
\text{HEAD} & \text{conj} \\
\text{IND} & s_0 \\
\text{RESTR} & \langle \text{ARGS} \langle s_1 \ldots s_n \rangle \rangle
\end{bmatrix}
\begin{bmatrix}
\text{FORM} & 1 \\
\text{VAL} & 0 \\
\text{IND} & s_n
\end{bmatrix}
\]
Lexical Types & Lexical Rules

• Lexemes capture the similarities among *run*, *runs*, *running*, and *ran*

• The lexical type hierarchy captures the similarities among *run*, *sleep*, and *laugh* (*siv-lxm*), among those and other verbs like *devour* and *hand* (*verb-lxm*), and among those and other words like *book* (*infl-lxm*).

• Lexical rules capture the similarities among *runs*, *sleeps*, *devours*, *hands*, ...
Three Kinds of Lexical Rules

• Inflectional: *lexeme* to *word*

• Derivational: *lexeme* to *lexeme*

• Post-Inflectional: *word* to *word*
  (Chapters 11, 13, 14)
Three Subtypes of $l$-rule

$l$-rule

$i$-rule  $d$-rule  $pi$-rule

\[ l$-rule : \]
\[
\begin{pmatrix}
\text{INPUT} & l$-sequence$\langle X , [ \text{SEM} / 2 ] \rangle \\
\text{OUTPUT} & l$-sequence$\langle Y , [ \text{SEM} / 2 ] \rangle \\
\end{pmatrix}
\]

$i$-rule :
\[
\begin{pmatrix}
\text{INPUT} & \langle X , \begin{bmatrix} \text{lexeme} \\ \text{SYN} & 3 \\ \text{ARG-ST} & A \end{bmatrix} \rangle \\
\text{OUTPUT} & \langle Y , \begin{bmatrix} \text{word} \\ \text{SYN} & 3 \\ \text{ARG-ST} & A \end{bmatrix} \rangle \\
\end{pmatrix}
\]

$d$-rule :
\[
\begin{pmatrix}
\text{INPUT} & \langle X , \begin{bmatrix} \text{lexeme} \\ \text{SYN} / 3 \\ \text{ARG-ST} \end{bmatrix} \rangle \\
\text{OUTPUT} & \langle Y , \begin{bmatrix} \text{lexeme} \\ \text{SYN} / 3 \end{bmatrix} \rangle \\
\end{pmatrix}
\]
\[ \text{i-rule} \]

**INPUT** \[ \left\langle 1, \text{cntn-lxm} \right\rangle \]

**OUTPUT** \[ \left\langle F_{NPL}(1), \begin{bmatrix} \text{word} \\ \text{SYN} \left[ \text{HEAD} \left[ \text{AGR} \left[ \text{NUM pl} \right] \right] \right] \end{bmatrix} \right\rangle \]
Plural Noun LR

\[
\begin{align*}
\text{INPUT} & \left\{ \begin{array}{c}
\text{i-rule} \\
\text{cntn-lxm}
\end{array} \right. \\
\text{OUTPUT} & \left\{ \begin{array}{c}
\text{word} \\
\text{F}_{NPL}(\Pi), \\
\text{HEAD} [\text{AGR} [\text{NUM pl}]]
\end{array} \right.
\end{align*}
\]
Plural Noun LR with Inherited Constraints

\[ \text{i-rule} \]

\[ \begin{align*}
\text{INPUT} & \quad \left\langle 1, \right. \\
& \quad \left. \begin{bmatrix}
\text{cntn-lxm} \\
\text{SEM} \\
\text{word}
\end{bmatrix} \right. \\
\text{OUTPUT} & \quad \left\langle F_{NPL}(1), \right. \\
& \quad \left. \begin{bmatrix}
\text{SEM} \\
\text{HEAD} [\text{AGR} [\text{NUM pl}]]
\end{bmatrix} \right. \\
& \quad \left. \begin{bmatrix}
\text{SEM} \\
\text{SEM}
\end{bmatrix} \right. \\
& \quad \left. \begin{bmatrix}
\text{SEM} \\
\text{SEM}
\end{bmatrix} \right. \\
& \quad \left. \begin{bmatrix}
\text{SEM} \\
\text{SEM}
\end{bmatrix} \right. \\
& \quad \left. \begin{bmatrix}
\text{SEM} \\
\text{SEM}
\end{bmatrix} \right. \\
\end{align*} \]
Plural Noun LR with Inherited Constraints

\[
\begin{align*}
\text{INPUT} & \quad \left< \begin{array}{c}
[1], \\
\text{SEM} & 2 \\
\text{ARG-ST} & B \oplus C
\end{array} \right> \\
\text{OUTPUT} & \quad \left< \begin{array}{c}
\text{F}_{NPL}(\Pi) , \\
\text{SEM} & 2 \\
\text{ARG-ST} & B \oplus C
\end{array} \right>
\end{align*}
\]
Plural Noun LR with Inherited Constraints

\[ i\text{-rule} \]

INPUT \( \langle 1 , \begin{cases} 
\text{SYN} & 3 \\
\text{SEM} & 2 \\
\text{ARG-ST} & B \oplus C 
\end{cases} \rangle \)

OUTPUT \( \langle F_{NPL}(\Pi) , \begin{cases} 
\text{SYN} & 3 \\
\text{SEM} & 2 \\
\text{ARG-ST} & B \oplus C 
\end{cases} \rangle \)
Plural Noun LR with Inherited Constraints

\[
\begin{aligned}
\text{INPUT} & \quad \left\langle 1, \begin{array}{c}
\text{SYN} \quad 3 \\
\text{VAL} \\
\text{SEM} \quad 2 \text{[MODE / ref]} \\
\text{ARG-ST} \quad B \oplus C
\end{array} \right\rangle \\
\text{OUTPUT} & \quad \left\langle F_{NPL}(1), \begin{array}{c}
\text{SYN} \quad 3 \\
\text{SEM} \quad 2 \\
\text{ARG-ST} \quad B \oplus C
\end{array} \right\rangle
\end{aligned}
\]
Plural Noun LR with Inherited Constraints

INPUT \( \left\langle 1, \begin{array}{c}
\text{SYN} \ 3 \\
\text{VAL} \\
\text{SEM} \\
\text{ARG-ST} \ B \oplus C
\end{array} \right\rangle \)

\( i \)-rule

OUTPUT \( \left\langle F_{NPL(2)}, \begin{array}{c}
\text{SYN} \ 3 \\
\text{VAL} \\
\text{SEM} \\
\text{ARG-ST} \ B \oplus C
\end{array} \right\rangle \)
Practicalities - Applying Lexical Rules

• INPUT is a family of lexical sequences.

• OUTPUT is another family of lexical sequences.
  • ...usually a smaller family
  • ...usually a disjoint one

• The only differences between the families are those stipulated in the rule (or the rule’s type).

• Similarities are handled by the constraints on \( l\text{-rule} \) and its subtypes.

• If we’ve written the LRs correctly, nothing is left underconstrained.
Example: Lexical Entry for \textit{cat}

\[
\langle \text{cat} , \left[ \begin{array}{c}
\text{cntn-lxm} \\
\text{SEM} \\
\text{RESTR} \\
\end{array} \right] \langle \left[ \begin{array}{c}
\text{INDEX } k \\
\text{RELN} \\
\text{INST } k \\
\end{array} \right] \rangle \rangle
\]
Example: *cat*, with inheritance
Licensing *cats*

INPUT: \[ \langle 1 \text{cat}, \rangle \]

\( i\text{-rule} \)

\( \text{cntn-lxm} \)

\( \text{SEM} \)

\( \text{ARG-ST} \)

OUTPUT: \[ \langle F_{\text{NPL}}(\text{II}), \rangle \]

\[ \langle \text{word}, \rangle \]

\[ \langle \text{HEAD}, \rangle \]

\[ \langle \text{VAL}, \rangle \]

\[ \langle \text{SEM}, \rangle \]

\[ \langle \text{ARG-ST}, \rangle \]
cats: The Lexical Sequence

\[
\begin{align*}
\text{word} &\rightarrow \text{HEAD} \left[ \text{noun} \right. \\
\text{SYN} &\rightarrow \text{VAL} \left[ \text{AGR} \ 3pl \right. \\
\text{SEM} &\rightarrow \text{INDEX} \left[ \text{ref} \right. \\
\text{ARG-ST} &\rightarrow \text{RESTR} \left[ \text{RELN} \ cat \right. \\
\end{align*}
\]
Derivational Lexical Rules

d-rule :

\[
\begin{bmatrix}
\text{INPUT} & \langle X, \left[ \text{lexeme SYN} \ / \ 3 \right] \rangle \\
\text{OUTPUT} & \langle Y, \left[ \text{lexeme SYN} \ / \ 3 \right] \rangle \\
\end{bmatrix}
\]
A Derivational LR: Agent Nominalization

\[
\begin{align*}
\text{INPUT} & \quad \langle 2, \left[ \begin{array}{c} \text{stv-lxm} \\ \text{SEM} \\ \text{ARG-ST} \langle X_i, \text{NP}_j \rangle \end{array} \right] \rangle \\
\text{OUTPUT} & \quad \langle F_{-er}(2), \left[ \begin{array}{c} \text{cntn-lxm} \\ \text{SEM} \\ \text{ARG-ST} \langle Y \left( \left[ \text{FORM of} \right], \text{PP}_j \right) \rangle \end{array} \right] \rangle
\end{align*}
\]

*the discoverer of oxygen*  
*a relier on Sandy*  
*a builder of bridges*  
*the putter of beer in the fridge*
\[
\langle \text{builder}, \quad \begin{array}{l}
\text{cntn-lxm} \\
\text{SYN} \quad \begin{array}{l}
\text{HEAD} \quad \left[ \text{noun} \right]
\text{AGR} \quad \left[ \text{PER} \quad \text{3rd} \right]
\end{array} \\
\text{VAL} \quad \left[ \text{SPR} \quad \left\langle \left[ \text{AGR} \quad \text{1} \right] \right\rangle \right]
\end{array}
\text{ARG-ST} \quad \left\langle \text{X}_i \quad \left( \text{PP} \left[ \text{of} \right]_j \right) \right\rangle
\end{array}
\text{MODE} \quad \text{ref}
\text{INDEX} \quad i
\right)
\text{SEM} \quad \begin{array}{l}
\text{RESTR} \quad \left\langle \left[ \text{RELN} \quad \text{build} \right] \right\rangle
\text{BUILDER} \quad i
\text{BUILT} \quad j
\end{array}
\text{ARG-ST} \quad \left\langle \left[ \text{COUNT} \quad + \right] \right\rangle
\]
Practicalities -- Writing Lexical Rules

- Determine the type of the LR.
- Determine the class of possible inputs.
- Determine what should change.
  - If INPUT and OUTPUT values are identified (by default or otherwise) and only OUTPUT value is mentioned, then... information is added.
    (Lexical sequences incompatible with that value are not possible inputs)
  - If INPUT and OUTPUT values are identified by default, but different values are given on the INPUT and OUTPUT of the rule, then... information is changed.
  - If INPUT and OUTPUT values are identified by an inviolable constraint, but different values are given on the INPUT and OUTPUT of the rule, then... there is no well-formed output.
Singular Noun Lexical Rule

\[ i\text{-rule} \]

INPUT \( \langle 1, \text{cn-lxm} \rangle \)

OUTPUT \( \langle 1, \left[ \text{SYN \ [HEAD \ [\text{AGR \ [\text{NUM sg}] outside]]}} \right] \rangle \)

Plural Noun Lexical Rule

\[ i\text{-rule} \]

INPUT \( \langle 1, \text{cntn-lxm} \rangle \)

OUTPUT \( \langle F_{\text{NPL}}(1), \left[ \text{SYN \ [HEAD \ [\text{AGR \ [\text{NUM pl}] outside]]}} \right] \rangle \)
3rd-Singular Verb Lexical Rule

\[
i-rule
\]

INPUT \[
\langle 3 , \left[ \begin{array}{c}
verb-lxm \\
SEM \\
RESTR \\
\end{array} \right] \rangle
\]

OUTPUT \[
\langle F_{3SG}(3) , \left[ \begin{array}{c}
SYN \\
HEAD \\
AGR \\
ARG-ST \\
\end{array} \right] \rangle
\]
Non-3rd-Singular Verb Lexical Rule

\[
i\text{-rule}
\]

\[
\text{INPUT} \quad \langle 1, \left[ \begin{array}{c}
\text{verb-lxm} \\
\text{SEM} [\text{RESTR } A]
\end{array} \right] \rangle
\]

\[
\text{OUTPUT} \quad \langle 1, \left[ \begin{array}{c}
\text{SYN} \\
\text{SEM} [\text{RESTR } A \oplus \ldots] \\
\text{ARG-ST} \langle \left[ \text{CASE nom} \right], \ldots \rangle
\end{array} \right] \rangle
\]
Past Tense Lexical Rule

\[
i \text{-rule}
\]

\[
\text{INPUT} \quad \left\langle 3, \begin{bmatrix} \text{verb-lxm} \\ \text{SEM} \begin{bmatrix} \text{RESTR} \ A \end{bmatrix} \end{bmatrix} \right\rangle
\]

\[
\text{OUTPUT} \quad \left\langle \text{F}_{PAST}(3), \begin{bmatrix} \text{SYN} \\ \text{SEM} \begin{bmatrix} \text{RESTR} \ A \oplus \ldots \end{bmatrix} \\ \text{ARG-ST} \begin{bmatrix} \text{CASE} \ nom \end{bmatrix}, \ldots \end{bmatrix} \right\rangle
\]
Base Form Lexical Rule

\[
\begin{align*}
\text{i-rule} \\
\text{INPUT} & \quad \langle 1, \text{verb-lxm} \rangle \\
\text{OUTPUT} & \quad \langle 1, \left[ \text{SYN} \left[ \text{HEAD} \left[ \text{FORM base} \right] \right] \right] \rangle \\
\end{align*}
\]

Constant Lexememe Lexical Rule

\[
\begin{align*}
\text{i-rule} \\
\text{INPUT} & \quad \langle 1, \text{const-lxm} \rangle \\
\text{OUTPUT} & \quad \left[ \text{FIRST} \ [1] \right] \\
\end{align*}
\]
Present Participle Lexical Rule

\[
\begin{align*}
\text{INPUT} & \quad \langle 3, \left[ \begin{array}{c}
\text{SEM} \\
\text{ARG-ST}
\end{array} \right] \left[ \text{RESTR A} \right] \rangle \\
\text{OUTPUT} & \quad \left[ F_{PRP}(3), \left[ \begin{array}{c}
\text{SYN} \\
\text{SEM} \\
\text{ARG-ST}
\end{array} \right] \left[ \text{HEAD} \left[ \text{FORM} \ prp \right] \right] \left[ \text{RESTR A} \oplus \ldots \right] \right]\end{align*}
\]
Past Participle Lexical Rule

\[
\begin{align*}
\text{d-rule} & \\
\text{INPUT} & \left< 3, \begin{bmatrix} \text{verb-lxm} \\ \text{SEM} \quad \begin{bmatrix} \text{RESTR} \ A \end{bmatrix} \\ \text{ARG-ST} \ B \end{bmatrix} \right> \\
\text{OUTPUT} & \left< F_{PSP}(3), \begin{bmatrix} \text{part-lxm} \\ \text{SYN} \quad \begin{bmatrix} \text{HEAD} \quad \begin{bmatrix} \text{FORM} \quad \text{psp} \end{bmatrix} \end{bmatrix} \\ \text{SEM} \quad \begin{bmatrix} \text{RESTR} \ A \odot \ldots \end{bmatrix} \\ \text{ARG-ST} \ B \end{bmatrix} \right>
\end{align*}
\]
Leading Ideas

• Rich lexicalism: words drive phrasal construction.
• Words are information-rich.
• This information is organized by a new concept: lexicon as a lexical database.
• “Vertical” generalizations expressed by lexical type hierarchy.
• “Horizontal” generalizations expressed by lexical rules (constructions).