JMORF/MORS — Morpho-Syntax

The Structure of the Lexicon

Francis Bond
Palacký University
https://fcbond.github.io/
bond@ieee.org

Lecture 7
Location: KB-1.11

Overview

- Lexical types: the lexeme hierarchy and default inheritance
- Lexical rules: inflectional, derivational and post-inflectional
- Problem: arguments in Japanese

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 $\left[\text{ARG-ST } / \left\langle \text{NP} \right\rangle\right]$,	and one of its subtypes says $\left[\text{ARG-ST } \left\langle\right\rangle\right]$,	then the ARG-ST value of instances of the subtype is $\left\langle\right\rangle$.
If a type says $\left[\text{ARG-ST } \left\langle \text{NP} \right\rangle\right]$,	and one of its subtypes says $\left[\text{ARG-ST } \left\langle\right\rangle\right]$,	then this subtype can have no instances, since they would have to satisfy contradictory constraints.

Default Inheritance, More Technicalities

- If a type says $\left[\text{MOD } / \left\langle \text{S} \right\rangle\right]$, and one of its subtypes says $\left[\text{MOD } \left\langle \left[\text{SPR } \left\langle \text{NP} \right\rangle\right] \right\rangle\right]$, then the ARG-ST value of instances of the subtype is what?

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

- That is, default constraints are 'pushed down'

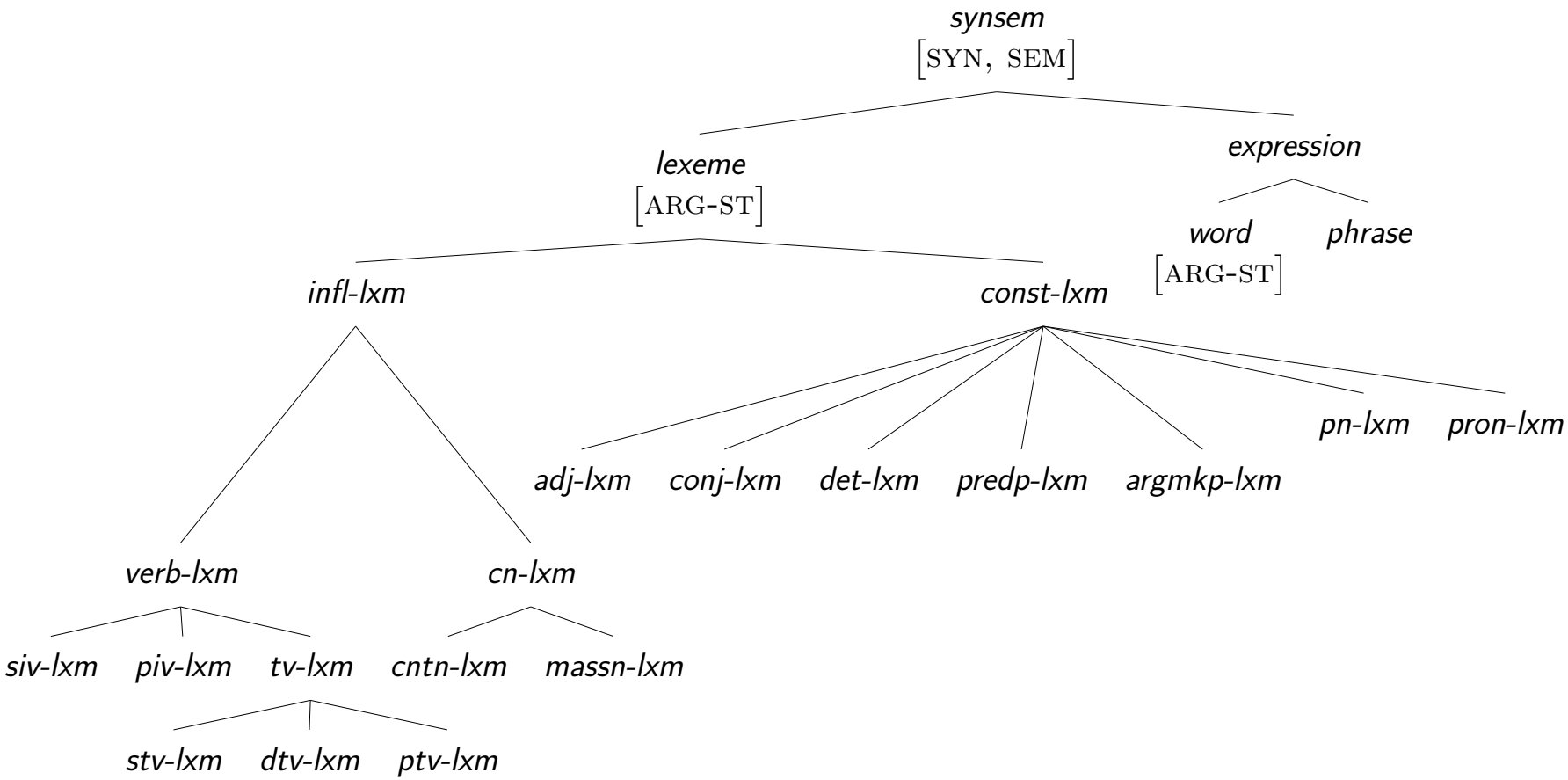
Question on Default Inheritance

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.

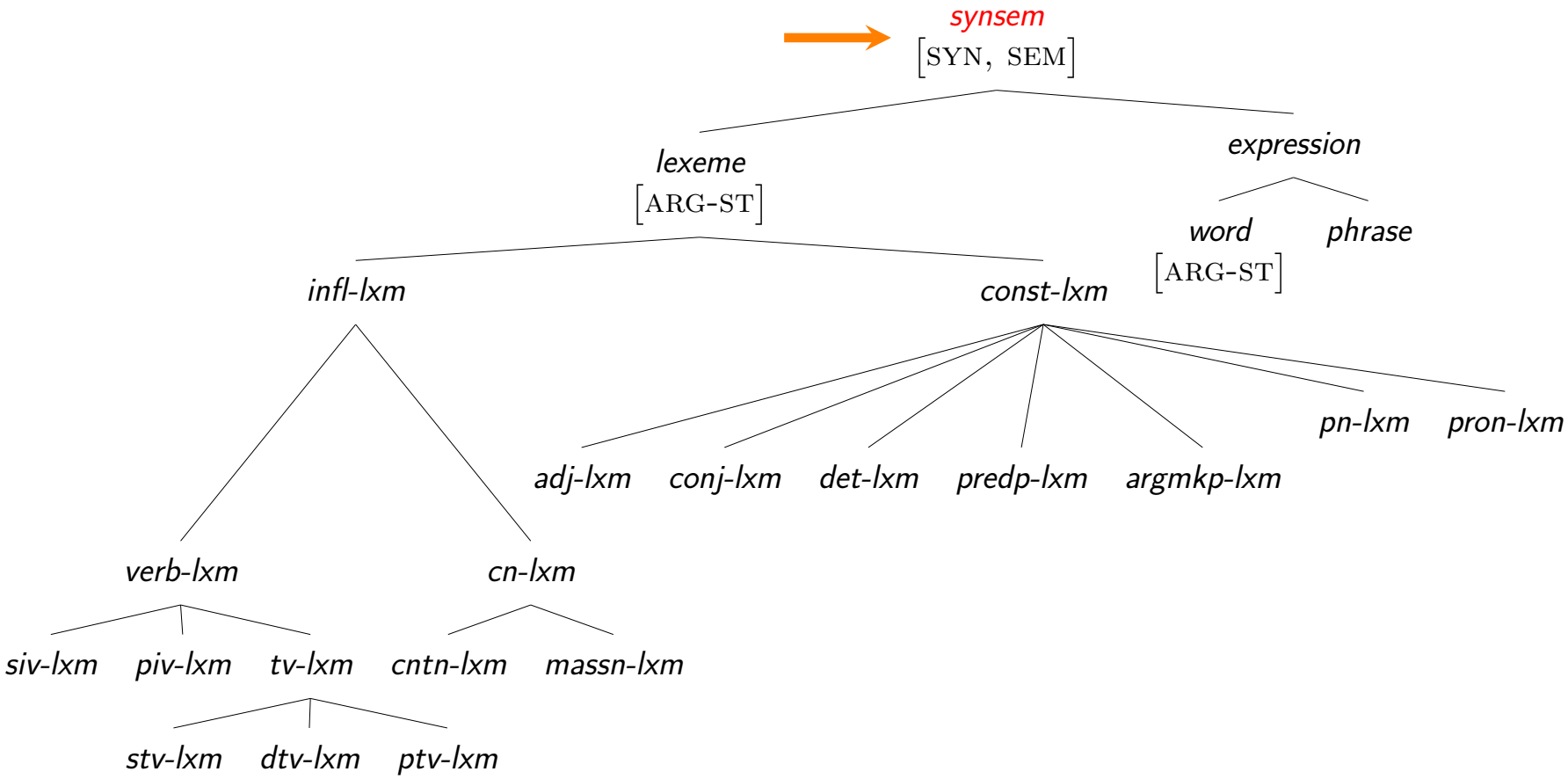
Our Lexeme Hierarchy



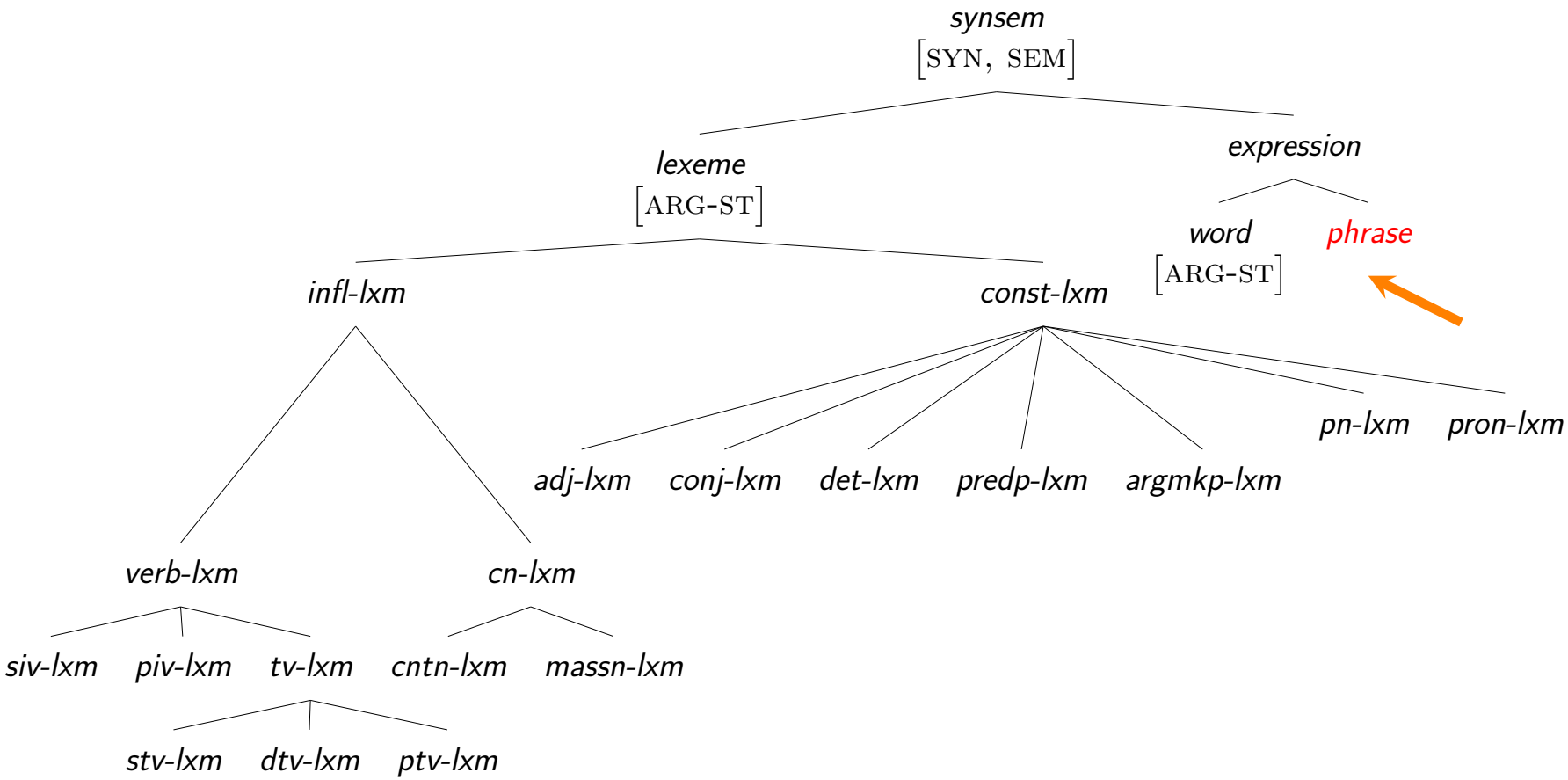
Functions of Types

- 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

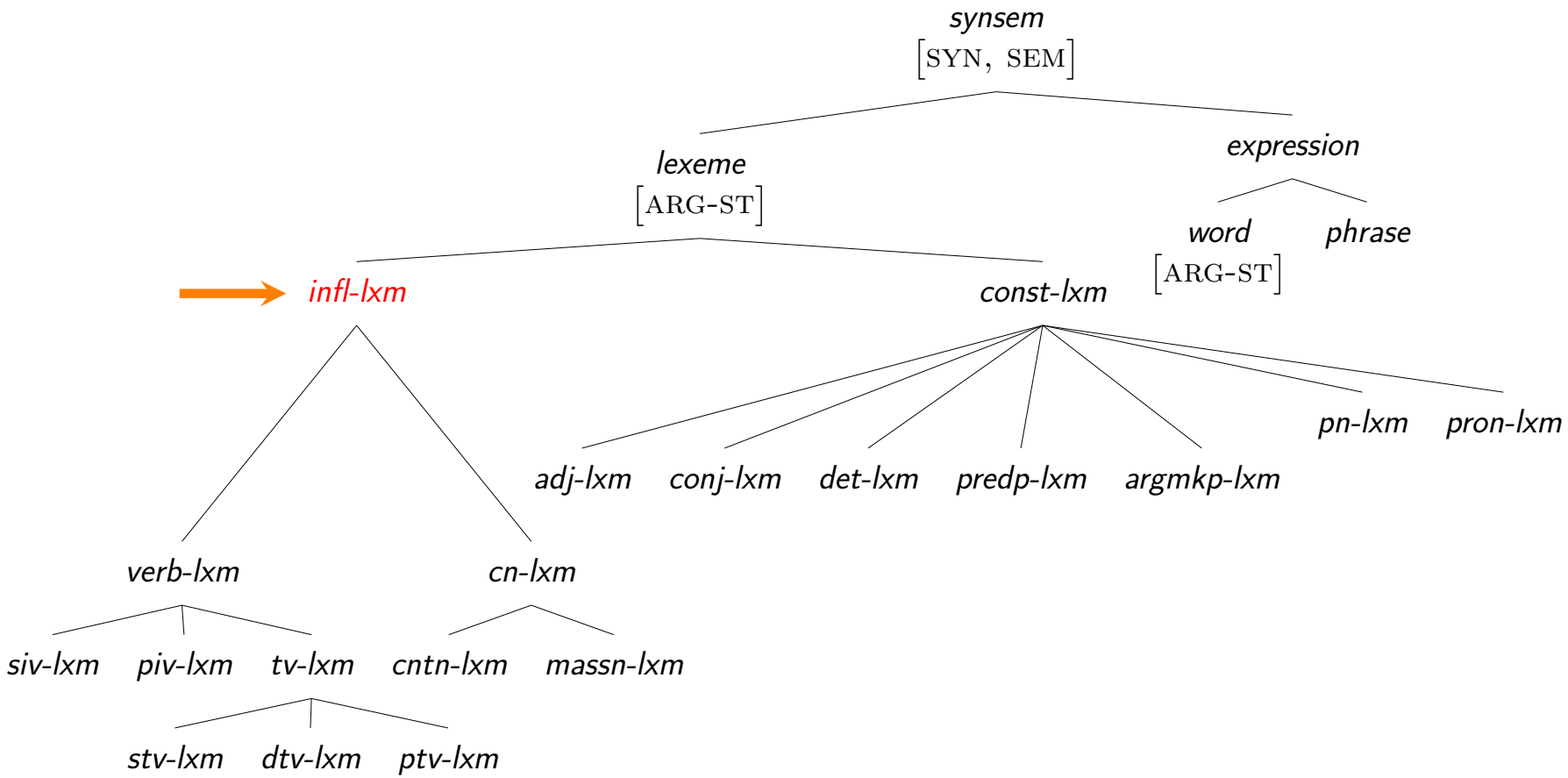
Every *synsem* has the Features SYN and SEM



No ARG-ST on *phrase*



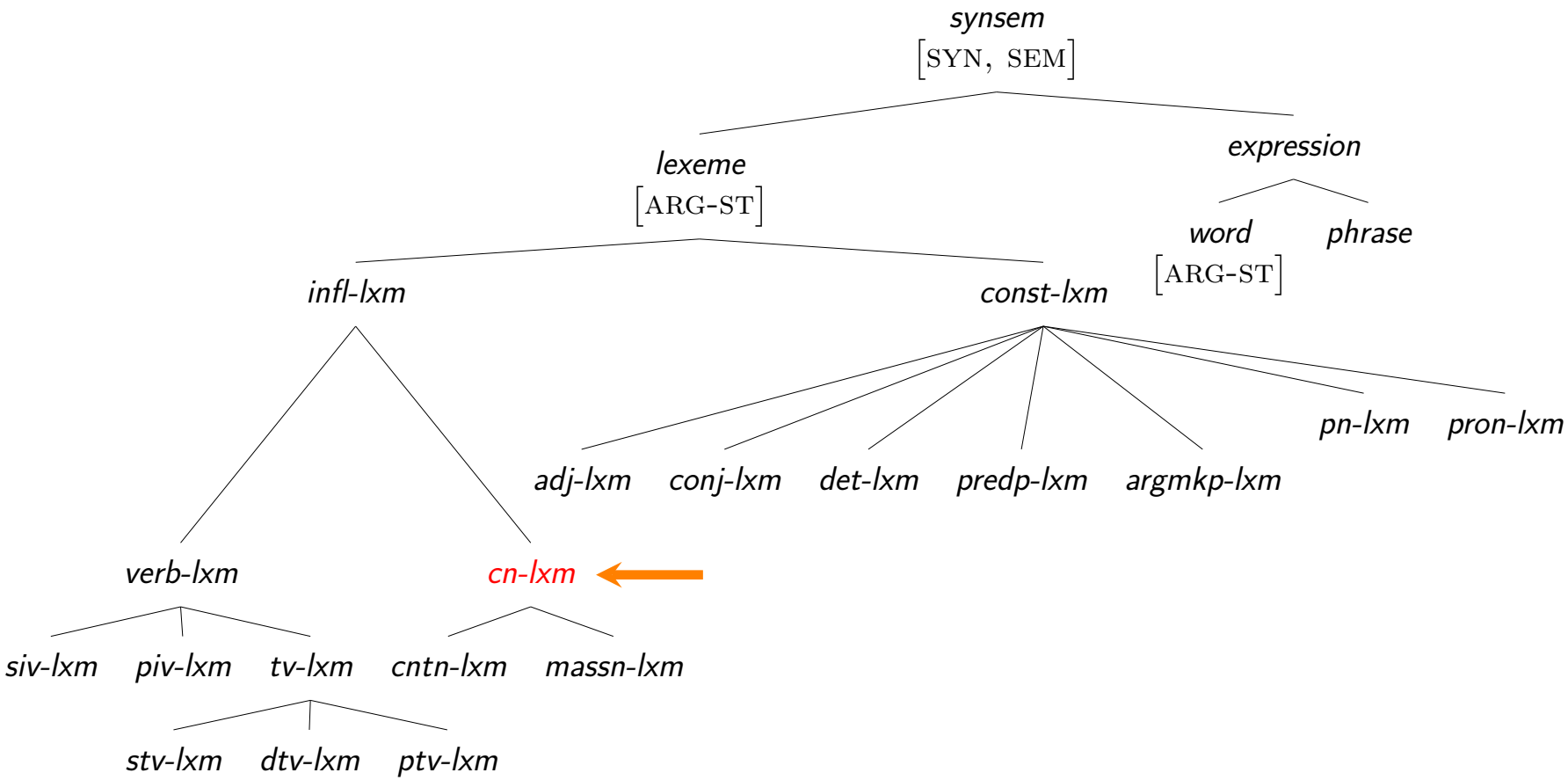
A Constraint on *infl-lxm*: the SHAC



A Constraint on *infl-lxm*: the SHAC

infl-lxm: $\left[\text{SYN} \left[\begin{array}{ll} \text{VAL} & \left[\text{SPR} \left\langle \left[\text{AGR } \boxed{1}\right] \right\rangle\right] \\ \text{HEAD} & \left[\text{AGR } \boxed{1}\right] \end{array}\right]\right]$

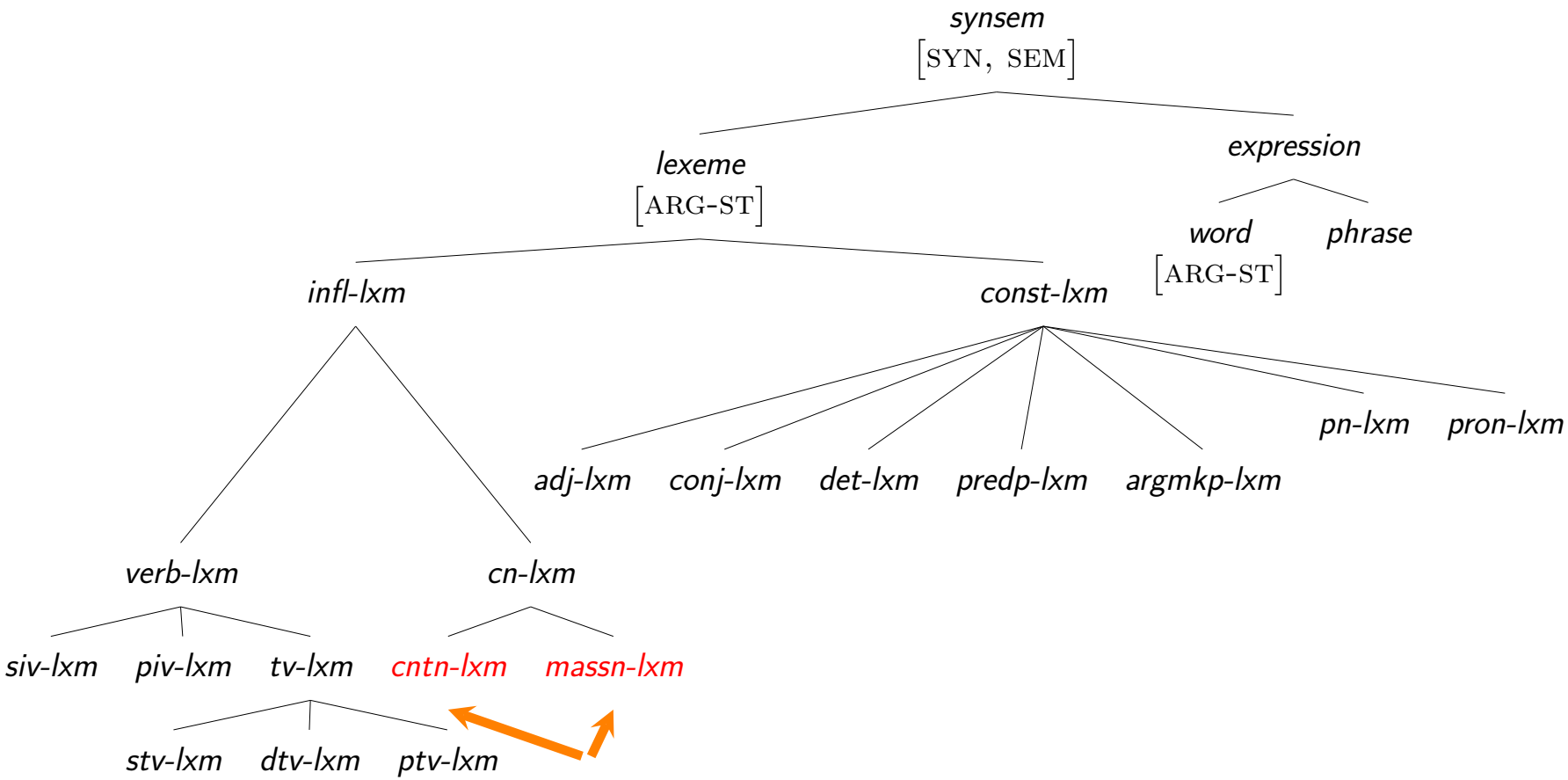
Constraints on *cn-lxm*



Constraints on *cn-lxm*

$$\textit{cn-lxm}: \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} \\ \text{AGR} & \begin{bmatrix} \text{PER} & \textit{3rd} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{bmatrix} \text{HEAD} & \textit{det} \\ \text{INDEX} & \textit{i} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & / & \textit{ref} \\ \text{INDEX} & & \textit{i} \end{bmatrix} \\ \text{ARG-ST} & \langle \text{X} \rangle \oplus & / & \langle\rangle \end{bmatrix}$$

Formally Distinguishing Count vs. Mass Nouns

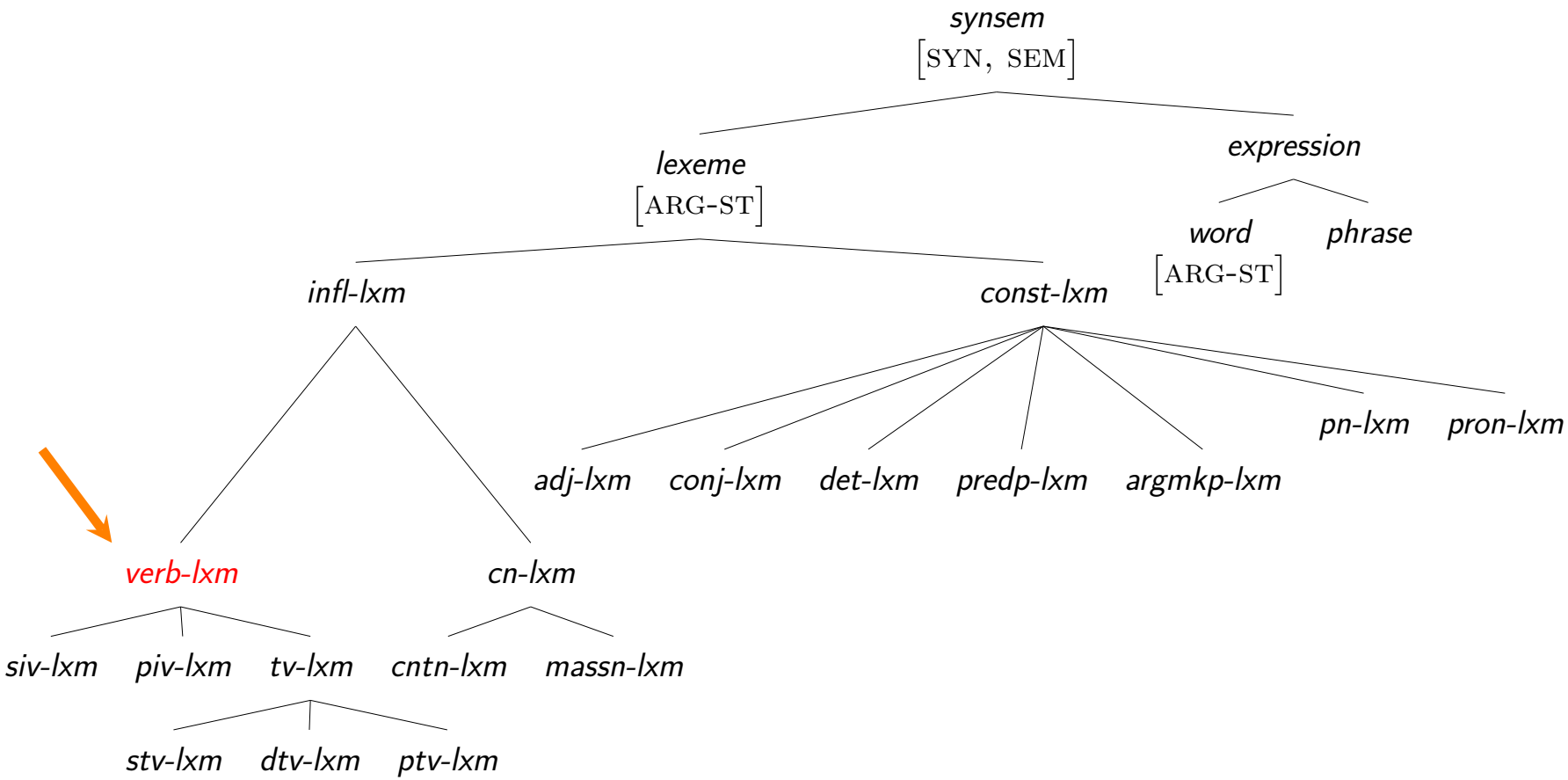


Formally Distinguishing Count vs. Mass Nouns

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

massn-lxm: $\left[\text{SYN}\ \left[\text{VAL}\ \left[\text{SPR}\ \left\langle\left[\text{COUNT}\ -\right]\right\rangle\right]\right]\right]$

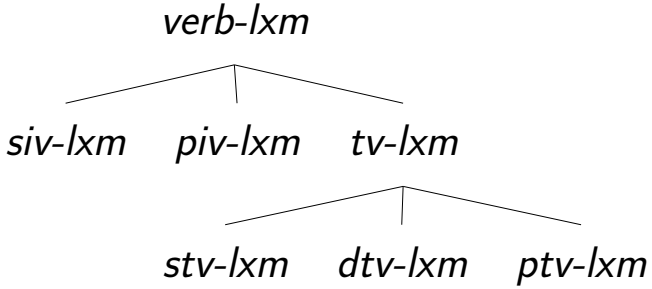
Constraints on *verb-lxm*



Constraints on *verb-lxm*

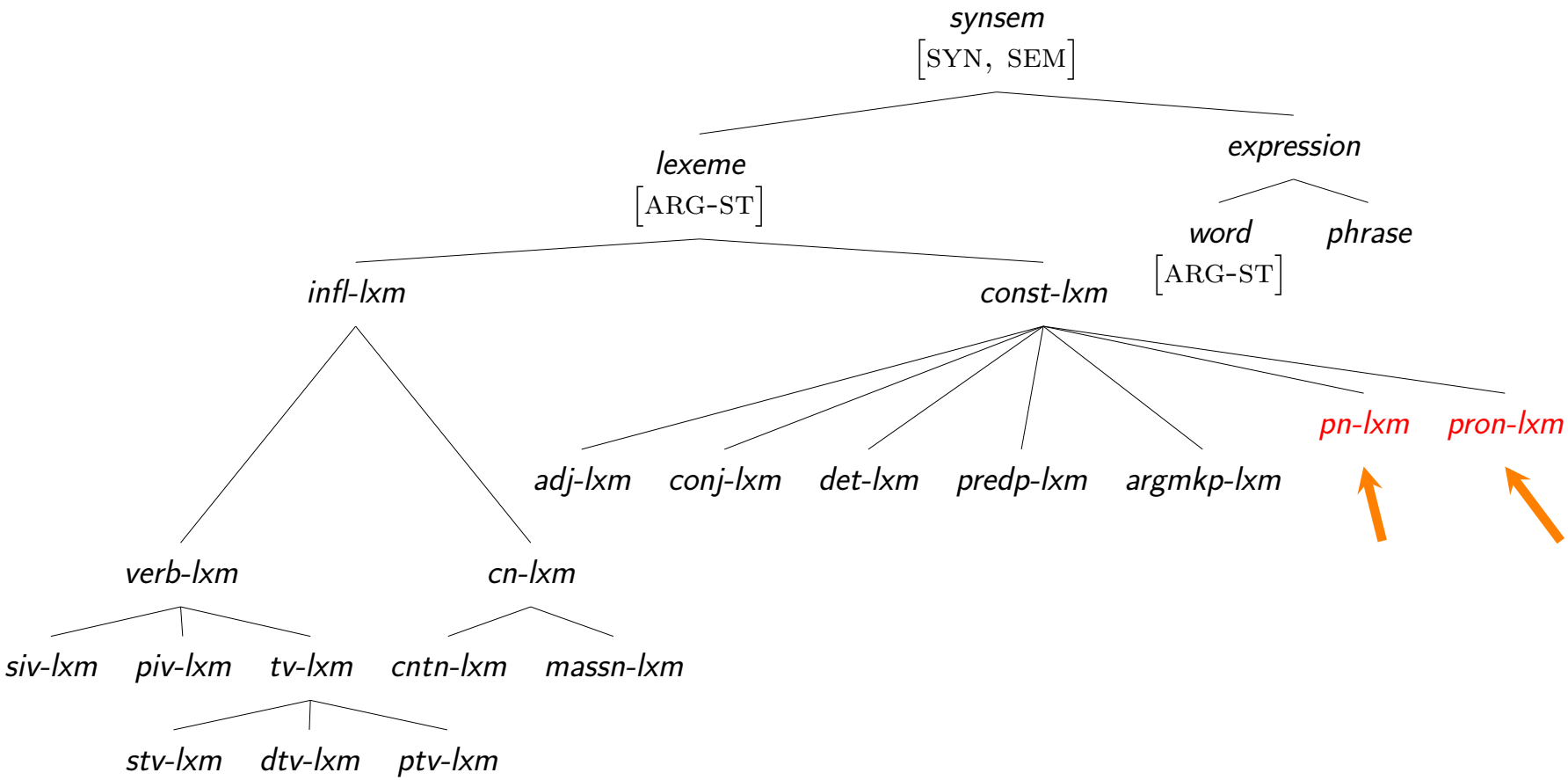
verb-lxm: $\begin{bmatrix} \textsc{syn} & \left[\textsc{head}\ \textit{verb}\right] \\ \textsc{sem} & \left[\textsc{mode}\ \textit{prop}\right] \\ \textsc{arg-st} & \left\langle \textsc{np},\ \ldots \right\rangle \end{bmatrix}$

Subtypes of *verb-lxm*



- *verb-lxm*: $\left[\textsc{arg-st} \; / \left\langle \text{NP}, \; \ldots \right\rangle\right]$
  - *siv-lxm*: $\left[\textsc{arg-st} \; / \left\langle \text{NP} \right\rangle\right]$
  - *piv-lxm*: $\left[\textsc{arg-st} \; / \left\langle \text{NP}, \; \text{PP} \right\rangle\right]$
  - *tv-lxm*: $\left[\textsc{arg-st} \; / \left\langle \text{NP}, \; \text{NP}, \; \ldots \right\rangle\right]$
    - *stv-lxm*: $\left[\textsc{arg-st} \; / \left\langle \text{NP}, \; \text{NP} \right\rangle\right]$
    - *dtv-lxm*: $\left[\textsc{arg-st} \; / \left\langle \text{NP}, \; \text{NP}, \; \text{NP} \right\rangle\right]$
    - *ptv-lxm*: $\left[\textsc{arg-st} \; / \left\langle \text{NP}, \; \text{NP}, \; \text{PP} \right\rangle\right]$

Proper Nouns and Pronouns



Proper Nouns and Pronouns

$$\textit{pn-lxm}: \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} \\ \text{AGR} & \begin{bmatrix} \text{PER} & \textit{3rd} \\ \text{NUM} & \textit{/ sg} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \textit{ref} \end{bmatrix} \\ \text{ARG-ST} & \textit{/} \langle\rangle \end{bmatrix}$$

$$\textit{pron-lxm}: \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{noun} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \textit{/ ref} \end{bmatrix} \\ \text{ARG-ST} & \langle\rangle \end{bmatrix}$$

The Case Constraint

An outranked NP is $\left[\textsc{case}\ \textit{acc}\right]$.

- object of verb ✓
- second object of verb ✓
- object of argument-marking preposition ✓
- object of predicational preposition (✓)

The Case Constraint, Continued

An outranked NP is $\left[\text{CASE } \textit{acc}\right]$.

- 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 them
  - 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*

Lexical Rules

Overview

- How lexical rules fit in
- Three types of lexical rules, constraints
- Example: Plural noun lexical rule
- Advice on writing lexical rules
- Constant lexemes
- ARG-ST & ARP
- The feature FORM

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*, 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*, ...

Parsimony & Plausibility

- Lexical rules capture **productive** generalizations.
- There may be some 'precompiling' going on as well.

Three Kinds of Lexical Rules

- **Inflectional**: *lexeme* to *word*

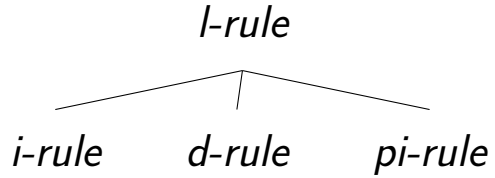
  Examples?

- **Derivational**: *lexeme* to *lexeme*

  Examples?

- **Post-Inflectional**: *word* to *word*
  (Chapters 11, 13, 14)

Three Subtypes of *l-rule*



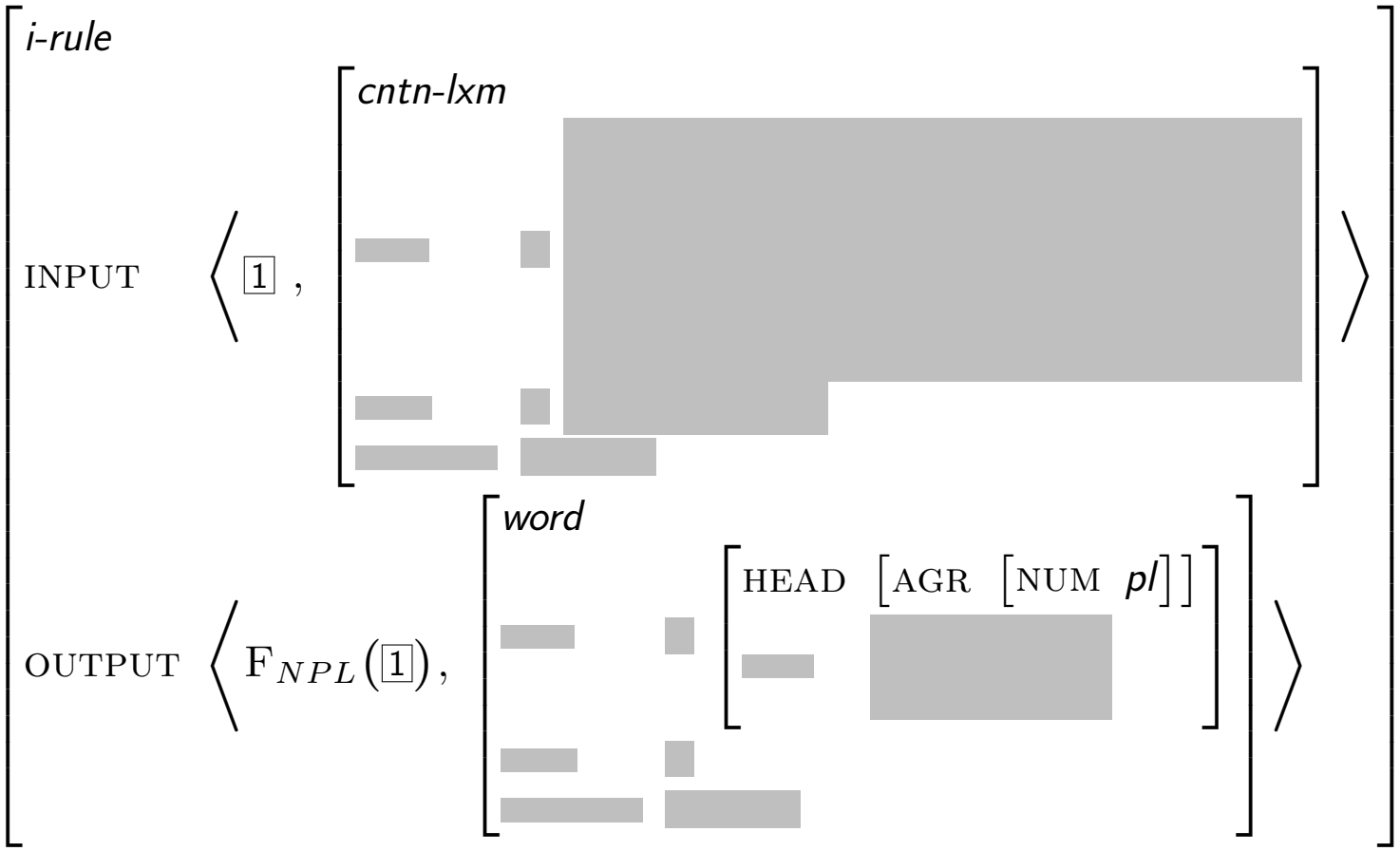
$$\textit{l-rule}: \begin{bmatrix} \textsc{input} & \textit{l-sequence} \left\langle \textsc{x} \ , \ \begin{bmatrix} \textsc{sem} \ / \boxed{2} \end{bmatrix} \right\rangle \\ \textsc{output} & \textit{l-sequence} \left\langle \textsc{y} \ , \ \begin{bmatrix} \textsc{sem} \ / \boxed{2} \end{bmatrix} \right\rangle \end{bmatrix}$$

$$\textit{i-rule}: \begin{bmatrix} \textsc{input} & \left\langle \textsc{x} \ , \ \begin{bmatrix} \textit{lexeme} \\ \textsc{syn} & \boxed{3} \\ \textsc{arg-st} & \boxed{A} \end{bmatrix} \right\rangle \\ \textsc{output} & \left\langle \textsc{y} \ , \ \begin{bmatrix} \textit{word} \\ \textsc{syn} & \boxed{3} \\ \textsc{arg-st} & \boxed{A} \end{bmatrix} \right\rangle \end{bmatrix} \qquad \textit{d-rule}: \begin{bmatrix} \textsc{input} & \left\langle \textsc{x} \ , \ \begin{bmatrix} \textit{lexeme} \\ \textsc{syn} \ / \boxed{3} \end{bmatrix} \right\rangle \\ \textsc{output} & \left\langle \textsc{y} \ , \ \begin{bmatrix} \textit{lexeme} \\ \textsc{syn} \ / \boxed{3} \end{bmatrix} \right\rangle \end{bmatrix}$$

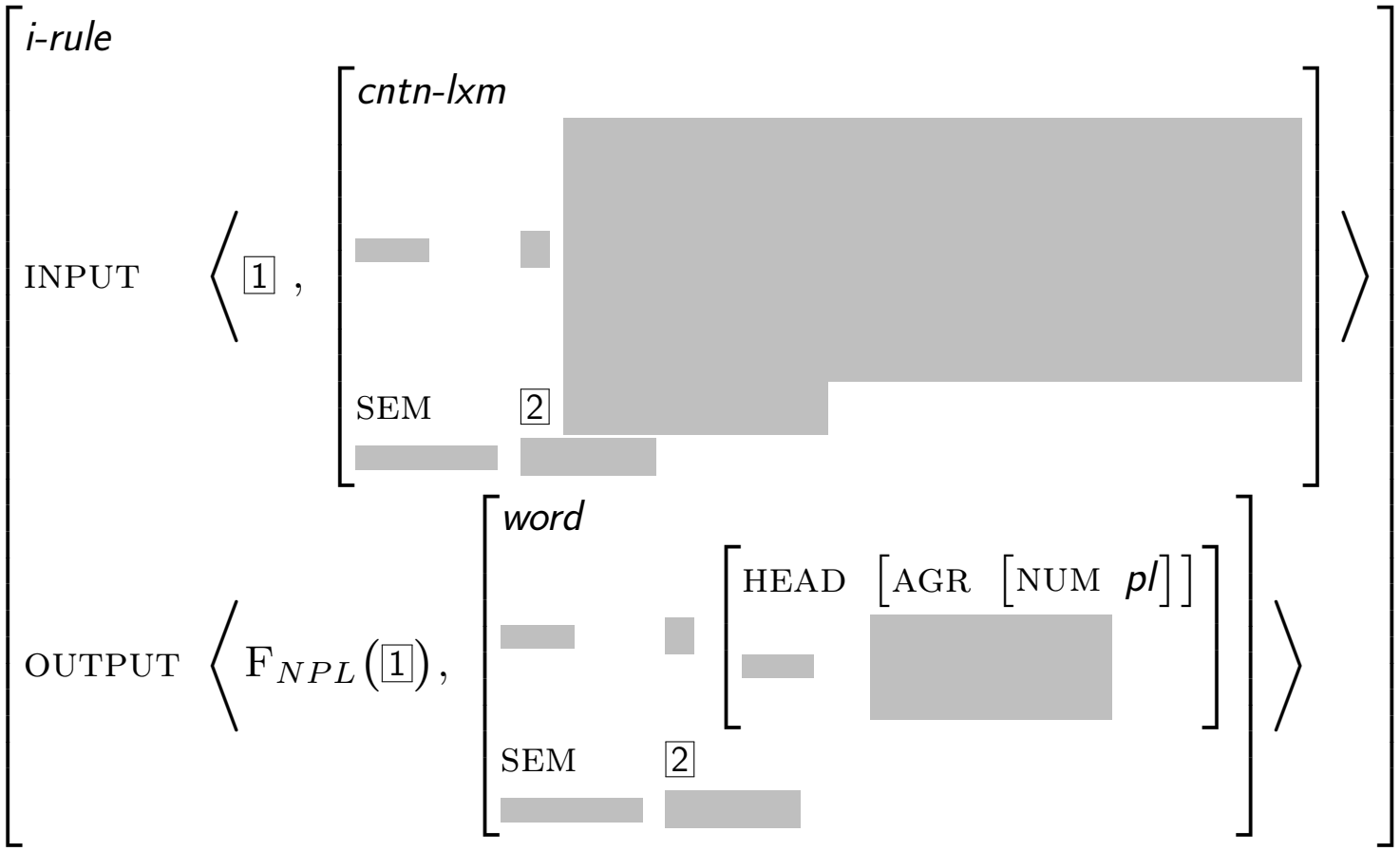
Plural Noun LR

$$\left[\begin{array}{l} \textit{i-rule} \\ \textsc{input} \quad \langle \boxed{1} \ , \ \textit{cntn-lxm} \rangle \\ \textsc{output} \quad \left\langle \mathrm{F}_{NPL}(\boxed{1}), \ \left[\begin{array}{l} \textit{word} \\ \textsc{syn} \ \left[\textsc{head} \ \left[\textsc{agr} \ \left[\textsc{num} \ \textit{pl}\right]\right]\right] \end{array}\right] \right\rangle \end{array}\right]$$

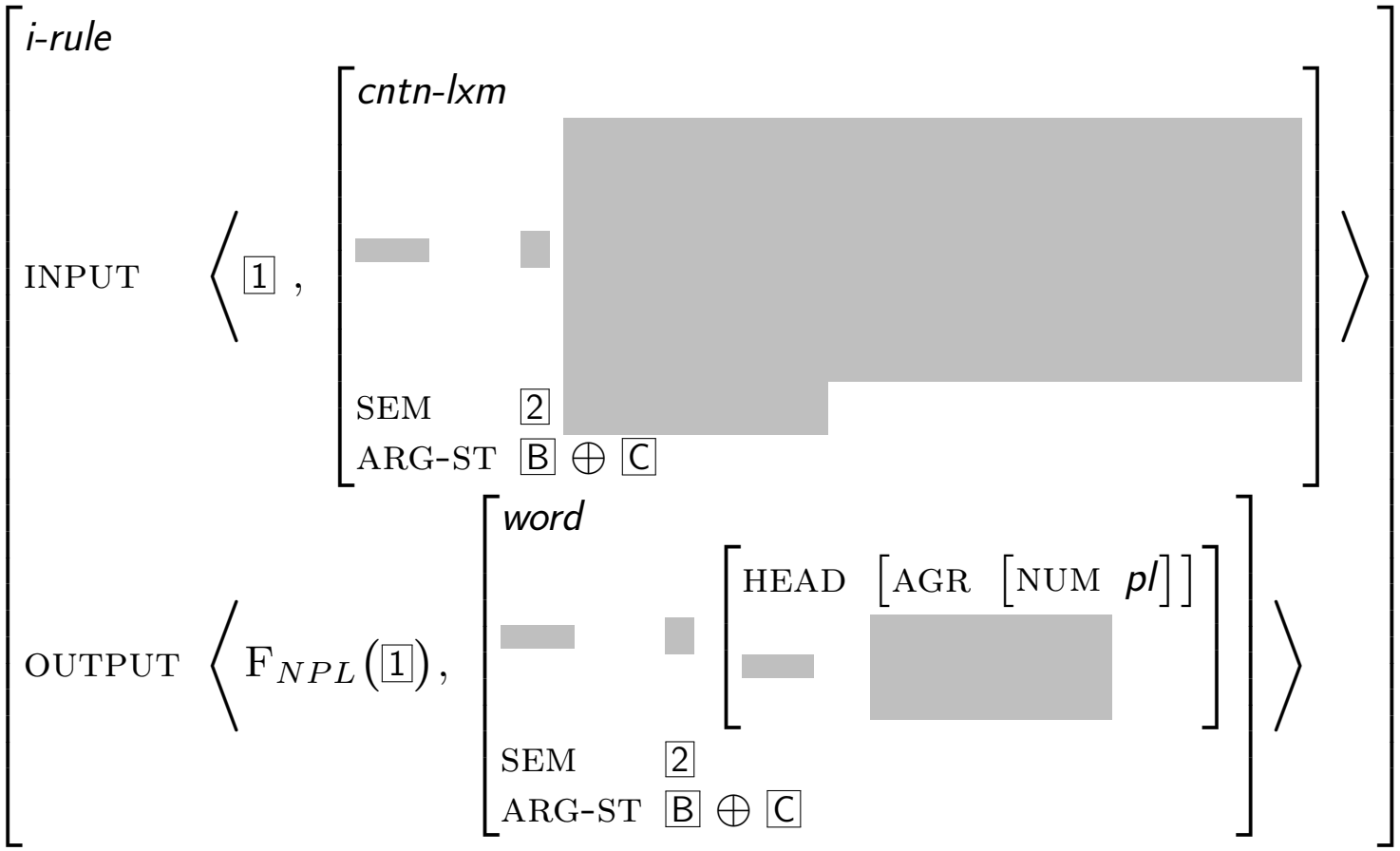
Plural Noun LR with Inherited Constraints



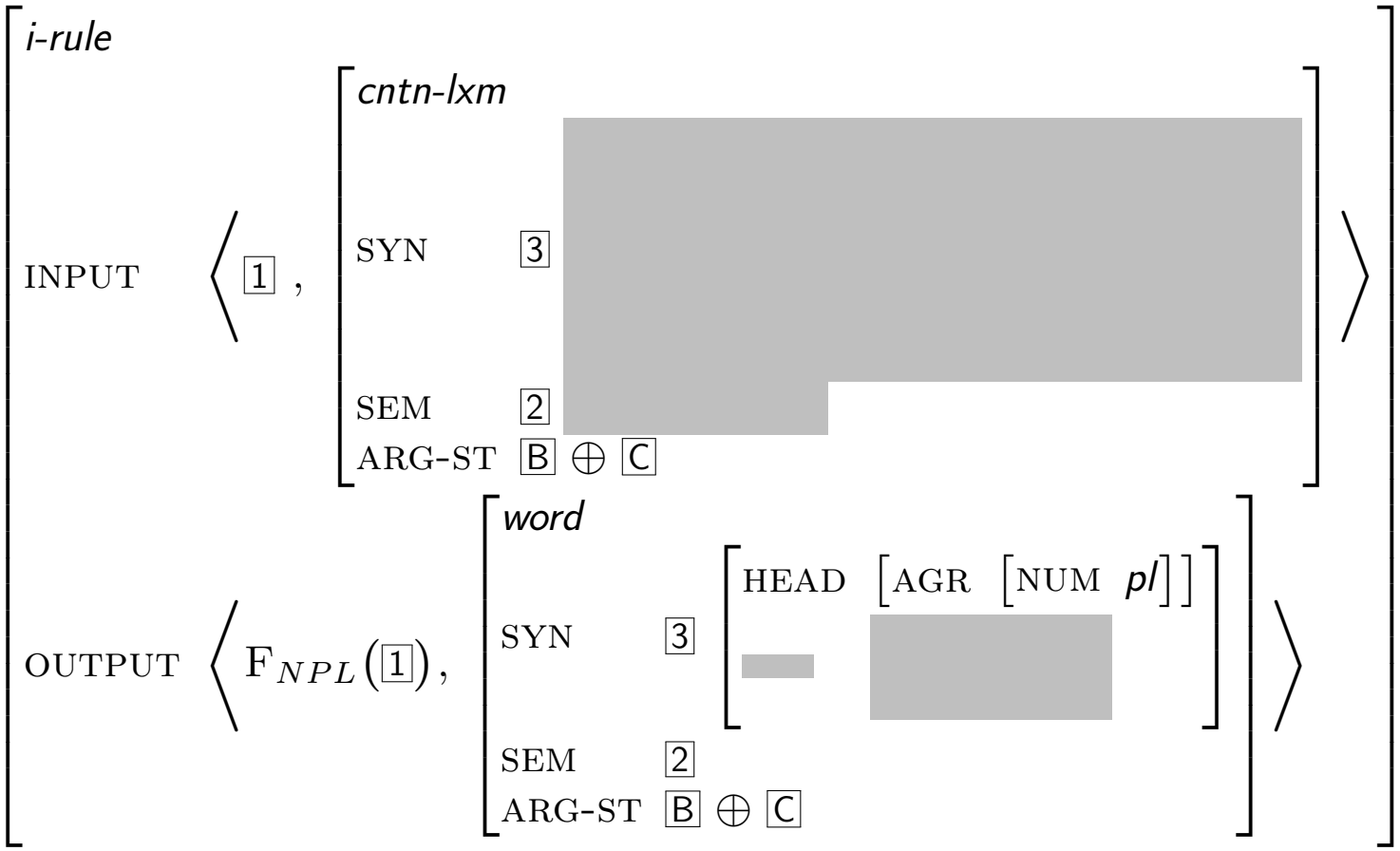
Plural Noun LR with Inherited Constraints



Plural Noun LR with Inherited Constraints



Plural Noun LR with Inherited Constraints



Plural Noun LR with Inherited Constraints

$$
\left[\begin{array}{l}
\textit{i-rule} \\
\textsc{input} \quad \left\langle \boxed{1}\,, \left[\begin{array}{l}
\textit{cntn-lxm} \\
\textsc{syn} \quad \boxed{3} \left[\begin{array}{ll}
\textsc{head} & \left[\begin{array}{l}\textit{noun} \\ \textsc{agr} \ \boxed{4}\left[\textsc{per} \ \textit{3rd}\right]\end{array}\right] \\
\textsc{val} & \left[\textsc{spr} \left\langle \textsc{dp} \left[\begin{array}{ll}\textsc{count} & + \\ \textsc{agr} & \boxed{4}\end{array}\right] \right\rangle\right]
\end{array}\right] \\
\textsc{sem} \quad \boxed{2}\left[\textsc{mode} \ / \ \textit{ref}\right] \\
\textsc{arg-st} \ \boxed{B} \oplus \boxed{C}
\end{array}\right] \right\rangle \\
\textsc{output} \quad \left\langle \mathrm{F}_{NPL}\left(\boxed{1}\right), \left[\begin{array}{l}
\textit{word} \\
\textsc{syn} \quad \boxed{3} \left[\begin{array}{ll}
\textsc{head} & \left[\textsc{agr} \ \left[\textsc{num} \ \textit{pl}\right]\right] \\
\blacksquare & \blacksquare
\end{array}\right] \\
\textsc{sem} \quad \boxed{2} \\
\textsc{arg-st} \ \boxed{B} \oplus \boxed{C}
\end{array}\right] \right\rangle
\end{array}\right]
$$

Plural Noun LR with Inherited Constraints

$$\left[\begin{array}{l} \textit{i-rule} \\ \text{INPUT} \quad \left\langle \boxed{1}, \left[\begin{array}{l} \textit{cntn-lxm} \\ \text{SYN} \quad \boxed{3} \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{l} \textit{noun} \\ \text{AGR} \ \boxed{4} \left[\text{PER} \ \textit{3rd}\right] \end{array}\right] \\ \text{VAL} & \left[\text{SPR} \left\langle \text{DP} \left[\begin{array}{ll} \text{COUNT} & + \\ \text{AGR} & \boxed{4} \end{array}\right] \right\rangle\right] \end{array}\right] \\ \text{SEM} \quad \boxed{2} \left[\text{MODE} \ / \ \textit{ref}\right] \\ \text{ARG-ST} \ \boxed{\text{B}} \oplus \boxed{\text{C}} \end{array}\right] \right\rangle \\ \text{OUTPUT} \quad \left\langle \text{F}_{NPL}\left(\boxed{1}\right), \left[\begin{array}{l} \textit{word} \\ \text{SYN} \quad \boxed{3} \left[\begin{array}{ll} \text{HEAD} & \left[\text{AGR} \ \left[\text{NUM} \ \textit{pl}\right]\right] \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \boxed{\text{B}} \\ \text{COMPS} & \boxed{\text{C}} \end{array}\right] \end{array}\right] \\ \text{SEM} \quad \boxed{2} \\ \text{ARG-ST} \quad \boxed{\text{B}} \oplus \boxed{\text{C}} \end{array}\right] \right\rangle \end{array}\right]$$

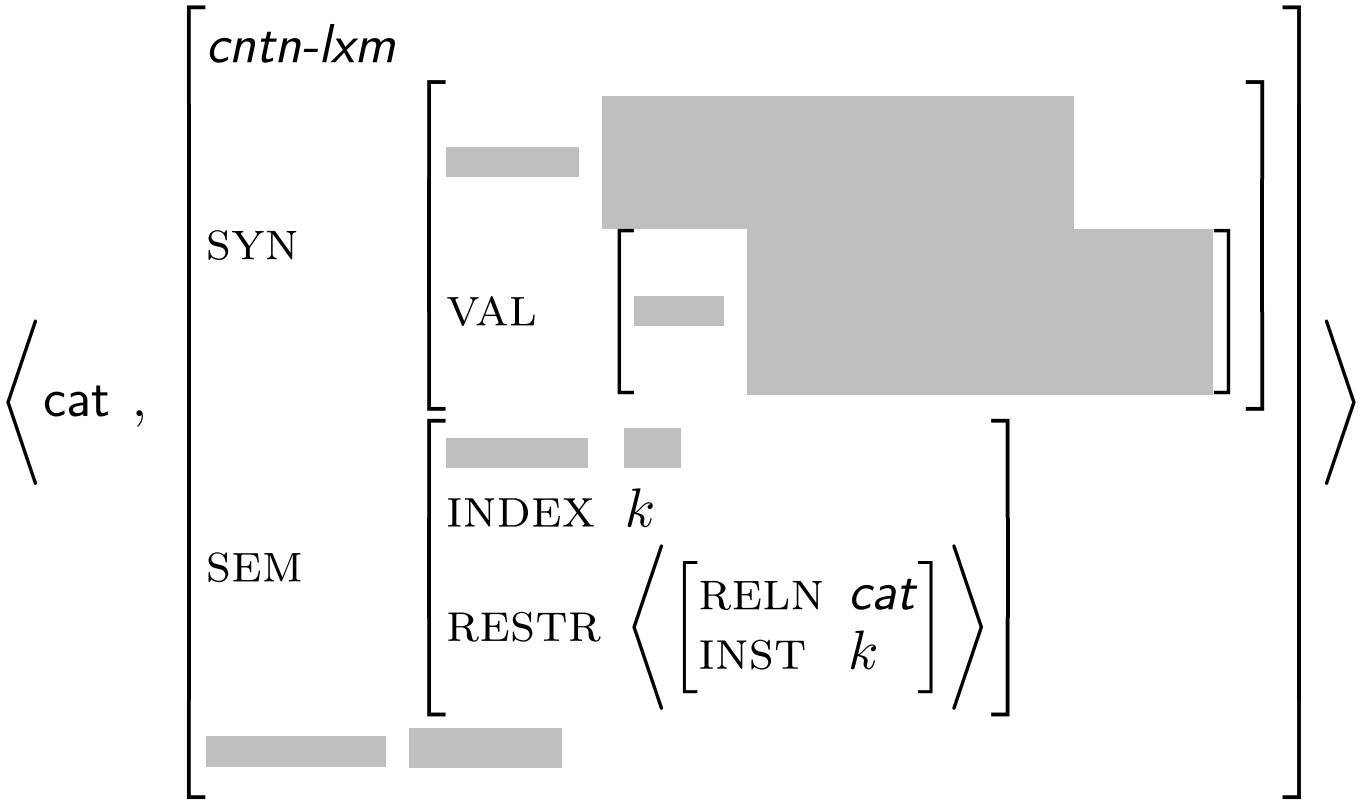
Practicalities — Applying Lexical Rules

- 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-rule* and its subtypes.
- If we've written the LRs correctly, nothing is left underconstrained.

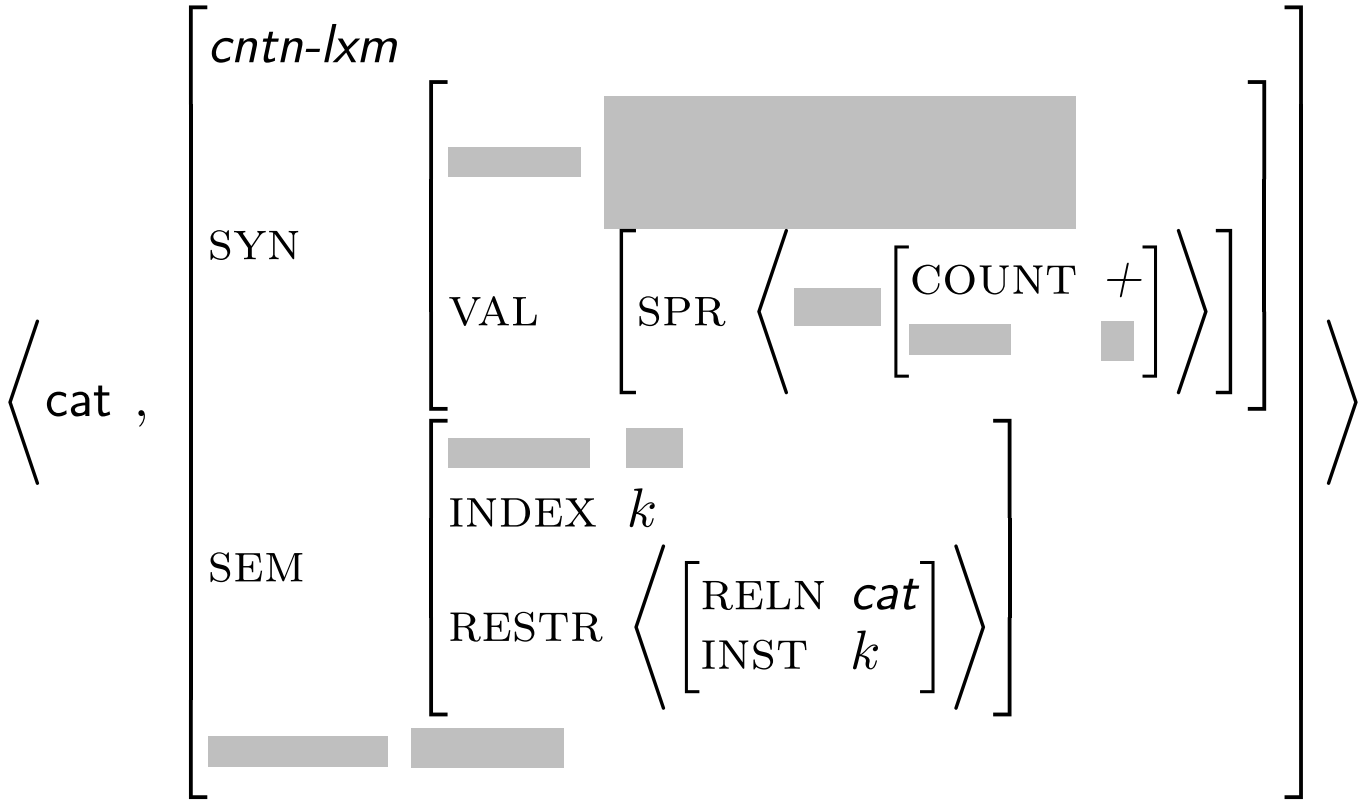
Example: Lexical Entry for *cat*

$$\left\langle \text{cat} \ , \begin{bmatrix} \textit{cntn-lxm} \\ \text{SEM} \begin{bmatrix} \text{INDEX} & k \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textit{cat} \\ \text{INST} & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Example: *cat*, with Inheritance



Example: *cat*, with Inheritance



Example: *cat*, with Inheritance

$$\left\langle \text{cat} \ , \begin{bmatrix} \textit{cntn-lxm} & & \\ \text{SYN} & \text{HEAD} & \begin{bmatrix} \textit{noun} \\ \text{AGR} \ \boxed{\ } \ \begin{bmatrix} \text{PER} & \textit{3rd} \end{bmatrix} \end{bmatrix} \\ & \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \text{DP} \begin{bmatrix} \text{COUNT} & + \\ \boxed{\ \ \ \ \ \ } & \boxed{\ } \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \textit{ref} \\ \text{INDEX} & k \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textit{cat} \\ \text{INST} & k \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{ARG-ST} & \langle \text{X} \rangle \end{bmatrix} \right\rangle$$

Example: *cat*, with Inheritance

$$\left\langle \text{cat} \ , \begin{bmatrix} \textit{cntn-lxm} & & \\ \text{SYN} & \text{HEAD} & \begin{bmatrix} \textit{noun} \\ \text{AGR} \ \boxed{7} \ \begin{bmatrix} \text{PER} \ \textit{3rd} \end{bmatrix} \end{bmatrix} \\ & \text{VAL} & \begin{bmatrix} \text{SPR} \ \left\langle \text{DP} \begin{bmatrix} \text{COUNT} \ + \\ \text{AGR} \ \ \boxed{7} \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} \ \ \textit{ref} \\ \text{INDEX} \ \ k \\ \text{RESTR} \ \left\langle \begin{bmatrix} \text{RELN} \ \ \textit{cat} \\ \text{INST} \ \ k \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{ARG-ST} & \langle \text{X} \rangle \end{bmatrix} \right\rangle$$

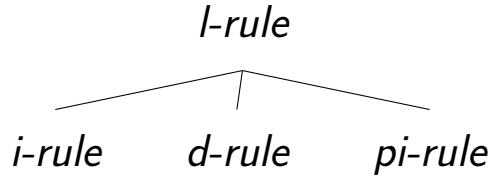
Plural Noun LR

$$\left[\begin{array}{l} \textit{i-rule} \\ \textsc{input} \quad \left\langle \boxed{1} \ , \ \textit{cntn-lxm} \right\rangle \\ \textsc{output} \quad \left\langle \mathrm{F}_{NPL}\left(\boxed{1}\right), \left[\begin{array}{l} \textit{word} \\ \textsc{syn} \ \left[\textsc{head} \ \left[\textsc{agr} \ \left[\textsc{num} \ \textit{pl}\right]\right]\right] \end{array}\right] \right\rangle \end{array}\right]$$

Licensing *cats*

$$\left[\begin{array}{l}\textit{i-rule}\\ \textsc{input}\quad \left\langle \boxed{1}\textsc{cat}\ ,\ \left[\begin{array}{l}\textit{cntn-lxm}\\ \textsc{syn}\quad \boxed{3}\ \left[\begin{array}{ll}\textsc{head} & \left[\begin{array}{l}\textit{noun}\\ \textsc{agr}\ \boxed{7}\ \left[\textsc{per}\ \textit{3rd}\right]\end{array}\right]\\ \textsc{val} & \left[\textsc{spr}\ \left\langle \textsc{dp}\left[\begin{array}{ll}\textsc{count} & +\\ \textsc{agr} & \boxed{7}\end{array}\right]\right\rangle\right]\end{array}\right]\\ \textsc{sem}\quad \boxed{2}\ \left[\begin{array}{ll}\textsc{mode} & \textit{ref}\\ \textsc{index} & k\\ \textsc{restr} & \left\langle\left[\begin{array}{ll}\textsc{reln} & \textit{cat}\\ \textsc{inst} & k\end{array}\right]\right\rangle\end{array}\right]\\ \textsc{arg-st}\ \boxed{B}\ \langle \textsc{x}\rangle \oplus \boxed{C}\ \langle\rangle\end{array}\right]\right\rangle\\ \textsc{output}\quad \left\langle \mathrm{F}_{NPL}\left(\boxed{1}\right)\ ,\ \left[\begin{array}{l}\textit{word}\\ \textsc{syn}\quad \boxed{3}\ \left[\begin{array}{ll}\textsc{head} & \left[\textsc{agr}\ \left[\textsc{num}\ \textit{pl}\right]\right]\\ \textsc{val} & \left[\begin{array}{ll}\textsc{spr} & \boxed{B}\\ \textsc{comps} & \boxed{C}\end{array}\right]\end{array}\right]\\ \textsc{sem}\quad \boxed{2}\\ \textsc{arg-st}\ \boxed{B}\oplus\boxed{C}\end{array}\right]\right\rangle\end{array}\right]$$

Three Subtypes of *l-rule*



$$\textit{l-rule}: \begin{bmatrix} \textsc{input} & \textit{l-sequence} \left\langle \textsc{x} \ , \begin{bmatrix} \textsc{sem} \ / \boxed{2} \end{bmatrix} \right\rangle \\ \textsc{output} & \textit{l-sequence} \left\langle \textsc{y} \ , \begin{bmatrix} \textsc{sem} \ / \boxed{2} \end{bmatrix} \right\rangle \end{bmatrix}$$

$$\textit{i-rule}: \begin{bmatrix} \textsc{input} & \left\langle \textsc{x} \ , \begin{bmatrix} \textit{lexeme} \\ \textsc{syn} & \boxed{3} \\ \textsc{arg-st} & \boxed{A} \end{bmatrix} \right\rangle \\ \textsc{output} & \left\langle \textsc{y} \ , \begin{bmatrix} \textit{word} \\ \textsc{syn} & \boxed{3} \\ \textsc{arg-st} & \boxed{A} \end{bmatrix} \right\rangle \end{bmatrix}$$

$$\textit{d-rule}: \begin{bmatrix} \textsc{input} & \left\langle \textsc{x} \ , \begin{bmatrix} \textit{lexeme} \\ \textsc{syn} \ / \boxed{3} \end{bmatrix} \right\rangle \\ \textsc{output} & \left\langle \textsc{y} \ , \begin{bmatrix} \textit{lexeme} \\ \textsc{syn} \ / \boxed{3} \end{bmatrix} \right\rangle \end{bmatrix}$$

cats: The Lexical Sequence

$$\left\langle \text{cats} \;,\; \left[\begin{array}{ll} \textit{word} & \\ \textsc{syn} & \left[\begin{array}{ll} \textsc{head} & \left[\begin{array}{ll} \textit{noun} & \\ \textsc{agr} & \textit{3pl} \end{array}\right] \\ \textsc{val} & \left[\begin{array}{ll} \textsc{spr} & \boxed{\text{B}} \left\langle \text{DP} \left[\begin{array}{ll} \textsc{count} & + \\ \textsc{agr} & \boxed{7} \end{array}\right] \right\rangle \\ \textsc{comps} & \langle\rangle \end{array}\right] \end{array}\right] \\ \textsc{sem} & \left[\begin{array}{ll} \textsc{mode} & \textit{ref} \\ \textsc{index} & k \\ \textsc{restr} & \left\langle \left[\begin{array}{ll} \textsc{reln} & \textit{cat} \\ \textsc{inst} & k \end{array}\right] \right\rangle \end{array}\right] \\ \textsc{arg-st} & \boxed{\text{B}} \end{array}\right] \right\rangle$$

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

Constant Lexemes

- What kinds of words are constant lexemes in our grammar?
- Why do we need a rule for these words?
- What would be an alternative analysis?

Constant Lexeme LR

$$\begin{bmatrix} \textit{i-rule} & \\ \textsc{input} & \langle \boxed{1}\ ,\ \textit{const-lxm} \rangle \\ \textsc{output} & \left[\textsc{first}\ \boxed{1}\right] \end{bmatrix}$$

- What keeps this from applying to, say, verb lexemes?
- Why is this an *i-rule*?

arg-st & ARP

- Given the ARP, what do we need to specify about the valence properties of words?
- Why isn't the ARP a constraint on the type *lexeme*?

The Feature form

- Different inflected forms of verbs show up in different syntactic environments. Examples?

- These different forms are syntactically distinguished by the feature FORM, as assigned by lexical rules.

- FORM is also useful in our analyses of coordination and PP selection.

What Rules These Out?

(1) **Kim eat pizza.*

(2) **Kim seems to eats pizza.*

(3) **Dana helped Leslie pack and moved.*

(4) **Kim relies for Sandy.*

(5) **Dana walked and Kim.*

Overview

- How lexical rules fit in
- Three types of lexical rules, constraints
- Example: Plural noun lexical rule
- Advice on writing lexical rules
- Constant lexemes
- ARG-ST & ARP
- The feature FORM

Problem

P1: Arguments in Japanese

As noted in Chapter 2, Japanese word order differs from English in a number of ways, including the fact that it is a 'Subject-Object-Verb' (SOV) language. The next two slides give some relevant examples. In the glosses, 'NOM', 'ACC', and 'DAT' stand for nominative, accusative, and dative case, respectively. (Note that Japanese has one more case — dative — than English does. This doesn't have any important effects on the analysis; it merely requires that we posit one more possible value of CASE for Japanese than for English.)[1]

[1] The examples marked with '*' here are unacceptable with the indicated meanings. Some of these might be well-formed with some other meaning of no direct relevance; others might be well-formed with special intonation that we will ignore for present purposes.

P1: Arguments in Japanese — Data

(6) *Hitorino otoko-ga sono hon-o yonda.*
one man-NOM that book-ACC read.PST
'One man read that book.'
[cf. *Yonda hitorino otoko-ga sono hon-o.
*Hitorino otoko-ga yonda sono hon-o.
*Otoko-ga hitorino sono hon-o yonda.
*Hitorino otoko-ga hon-o sono yonda.
*Hitorino otoko-ni/-o sono hon-o yonda.
*Hitorino otoko-ga sono hon-ga/-ni yonda.]

P1: Arguments in Japanese — Data (continued)

(7) *Hanako-ga hon-o yonda.*
Hanako-NOM book-ACC read.PST
'Hanako read the book(s).'
[cf. *Yonda Hanako-ga hon-o.
*Hanako-ga yonda hon-o.
*Hanako-ni/-o hon-o yonda.
*Hanako-ga hon-ni/-ga yonda.]

P1: Arguments in Japanese — More Data

(8) *Sensei-ga Taroo-ni sono hon-o ageta.*
teacher-NOM Taroo-DAT that book-ACC give.PST
'The teacher(s) gave that book to Taroo.'
[cf. *Ageta sensei-ga Taroo-ni sono hon-o.
*Sensei-ga ageta Taroo-ni sono hon-o.
*Sensei-ga Taroo-ni ageta sono hon-o.
*Sensei-o/-ni Taroo-ni sono hon-o ageta.
*Sensei-ga Taroo-ga/-o sono hon-o ageta.
*Sensei-ga Taroo-ni sono hon-ga/-ni ageta.]

(9) *Hanako-ga kita.*
Hanako-NOM arrive.PST
'Hanako arrived.'
[cf. *Kita Hanako-ga.]

P1: Arguments in Japanese — Tasks

As the contrasting ungrammatical examples show, the verb must appear in final position in Japanese. In addition, we see that verbs select for NPs of a particular case, much as in English. In the following tasks, assume that the nouns and verbs of Japanese are inflected words, derived by lexical rule from the appropriate lexemes.

A. Write Head-Specifier and Head-Complement Rules for Japanese that account for the data illustrated here. How are they different (if at all) from the Head-Specifier and Head-Complement Rules for English?

P1: Arguments in Japanese — Tasks (continued)

B. Give the lexical entry for each of the verbs illustrated in (6)–(9).

   [*Make sure your entries interact with the rules you formulated in part (A) to account for the above data. The data given permit you to specify only some features; leave others unspecified. Assume that there is a Past-Tense Verb Lexical Rule (an i-rule) that relates your lexical entries to the words shown in (6)–(9). We have not provided a hierarchy of lexeme types for Japanese. You may either give all relevant constraints directly on the lexical entries, or posit and use subtypes of lexeme. In the latter case, you must also provide those types.*]

C. Give the lexical entries for the nouns *Taroo* and *hon*. [*See notes on part (B).*]

D. Formulate the lexical rule for deriving the inflected forms ending in *-o* from the nominal lexemes.

Acknowledgments and References

- Course design and slides borrow heavily from Emily Bender's course: *Linguistics 566: Introduction to Syntax for Computational Linguistics* http://courses.washington.edu/ling566

- The slides on lexical types and lexical rules are adapted from Emily Bender's, with her permission; the feature structures follow Sag et al. (2003), © 2003 CSLI Publications

- This lecture complements Sag et al. (2003, Chapter 8)

References

Ivan A. Sag, Tom Wasow, and Emily Bender. 2003. *Syntactic Theory: A Formal Introduction*. CSLI Publications, Stanford, second edition.