

Ling 566
Oct 28, 2010
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.

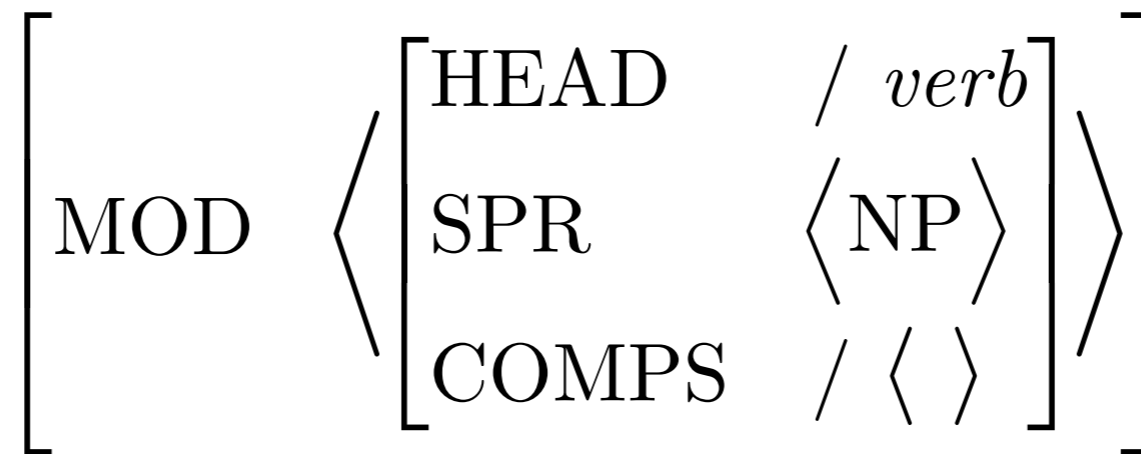
Default Inheritance, Technicalities

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



- That is, default constraints are ‘pushed down’

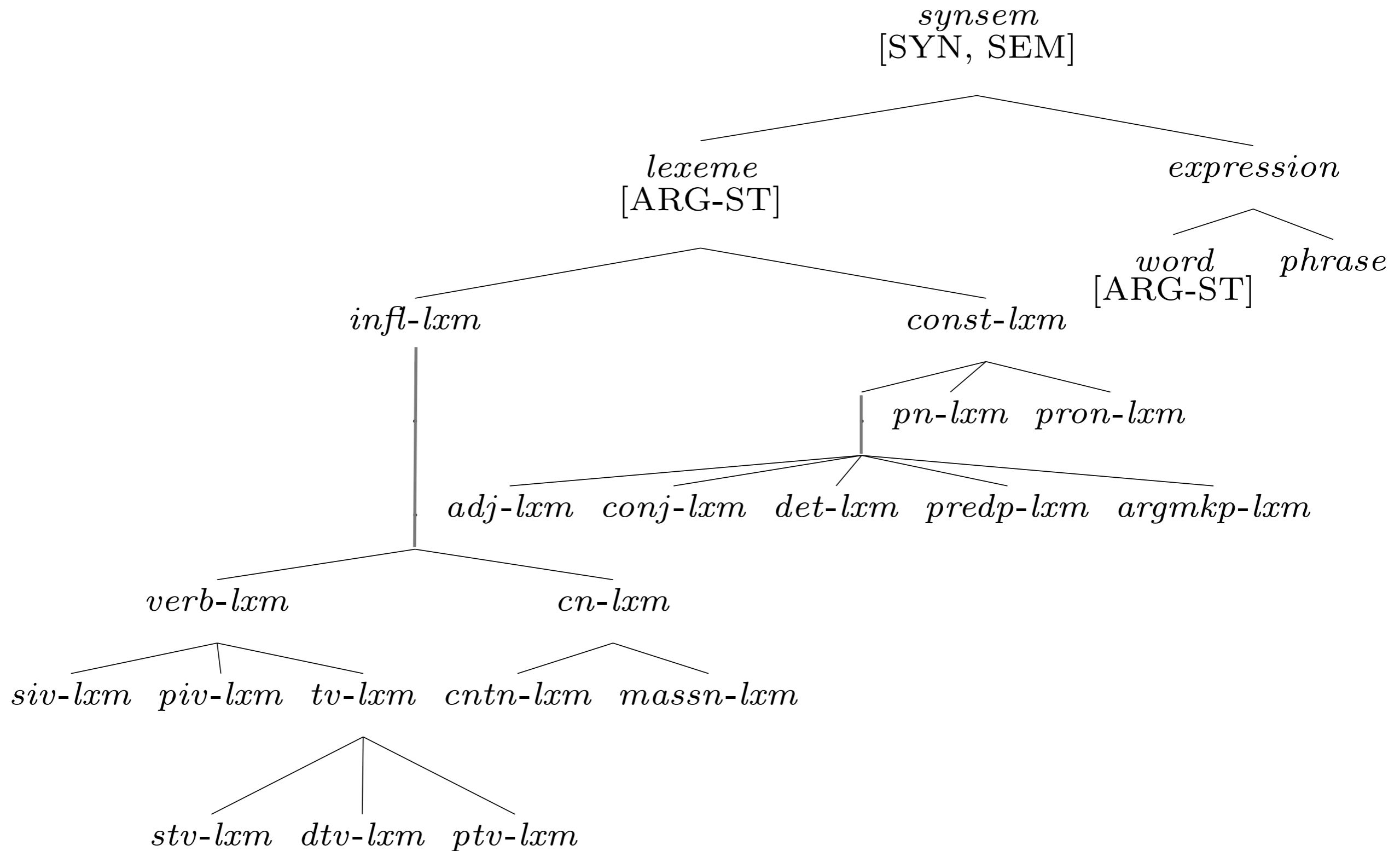
Question on Default Inheritance

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.

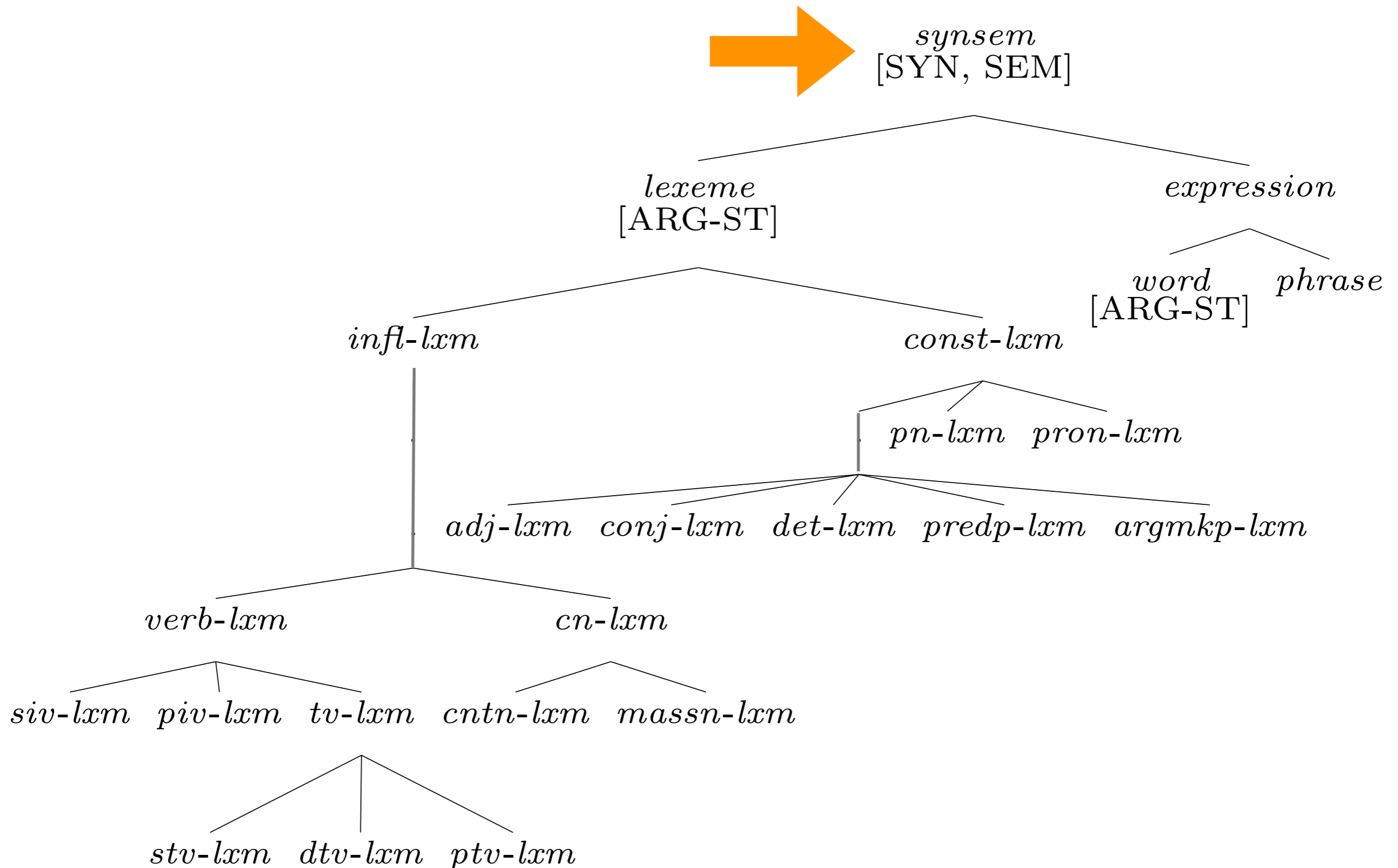
Our Lexeme Hierarchy



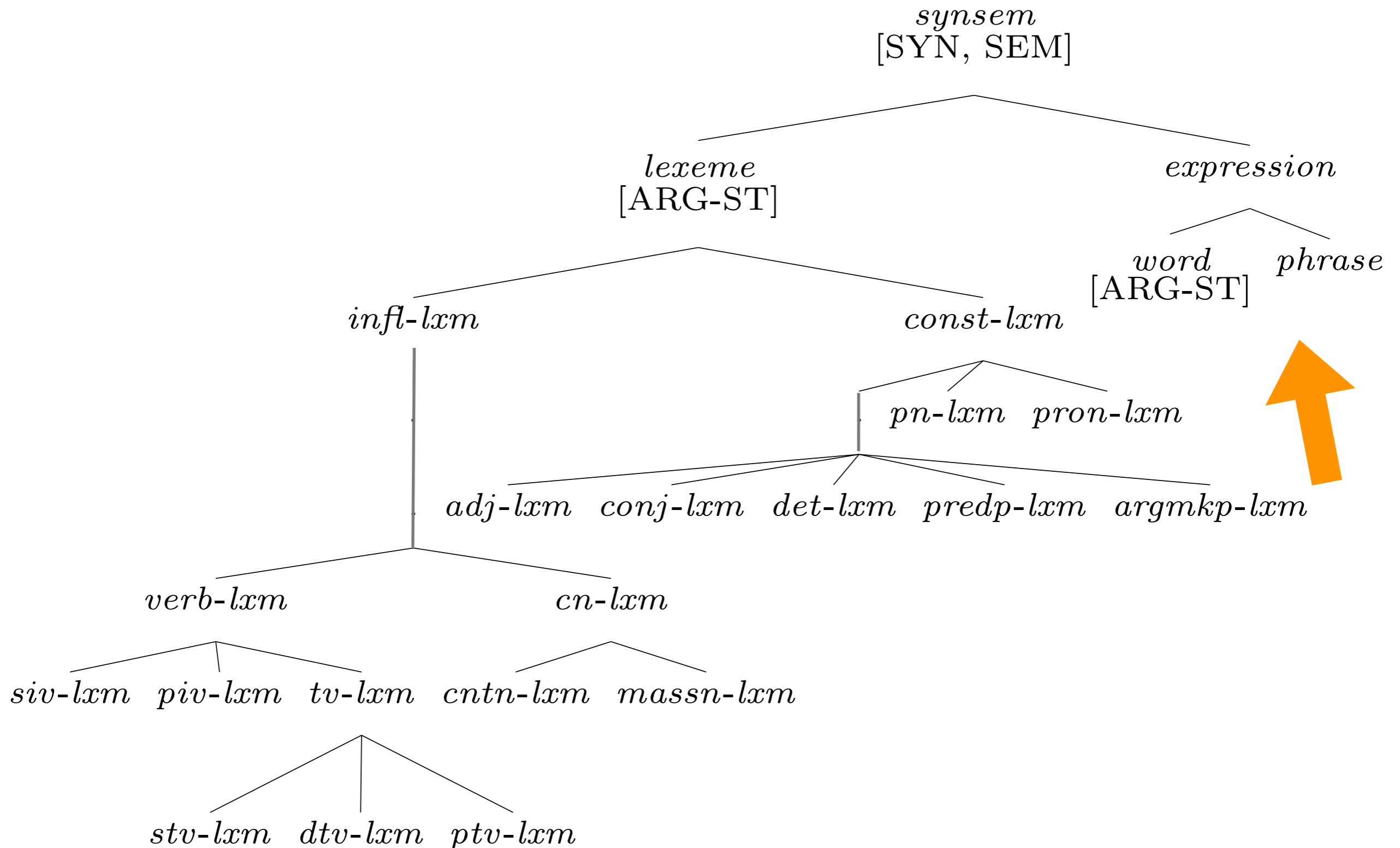
Functions of Types

- 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

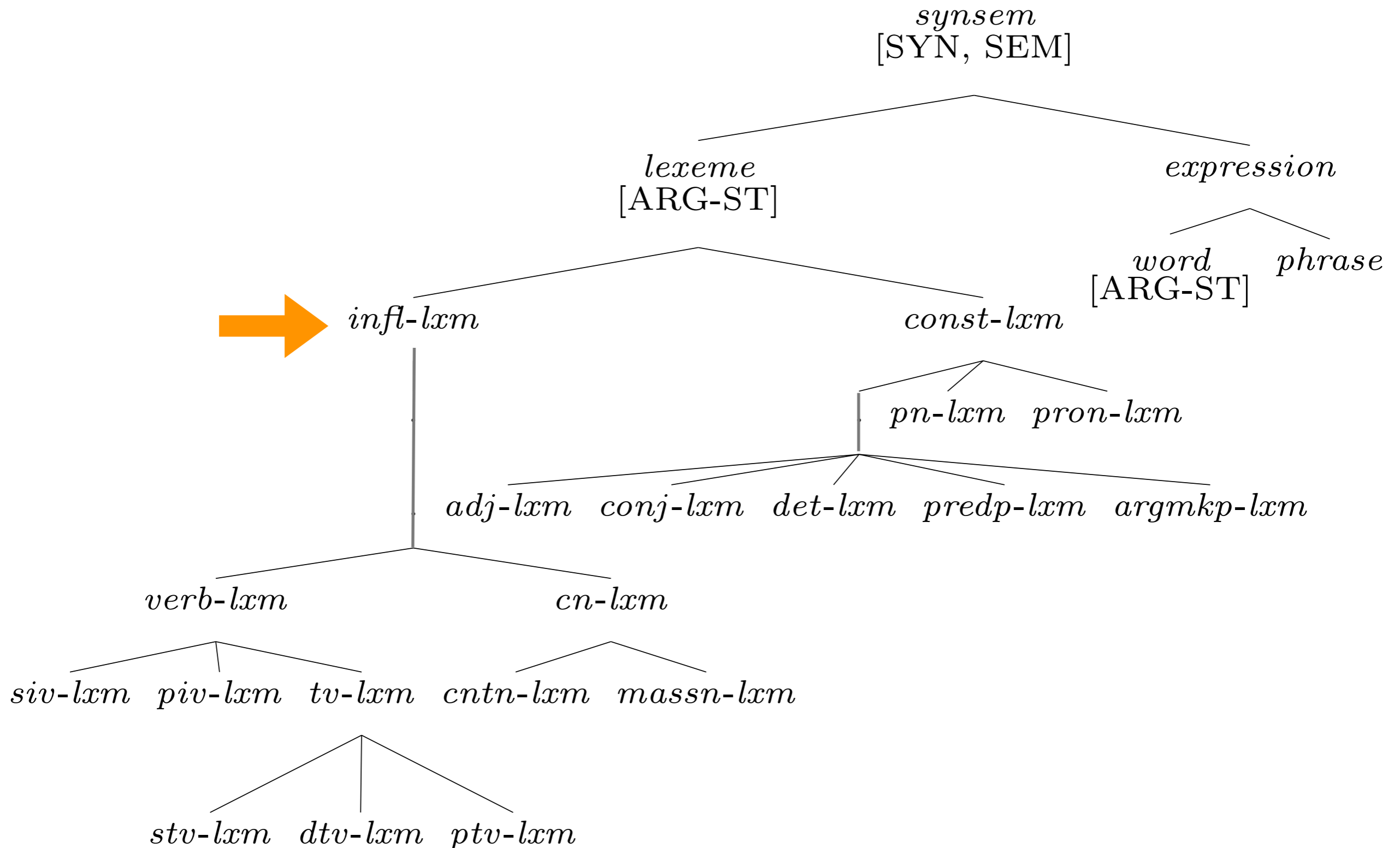
Every *synsem* has the features SYN and SEM



No ARG-ST on *phrase*



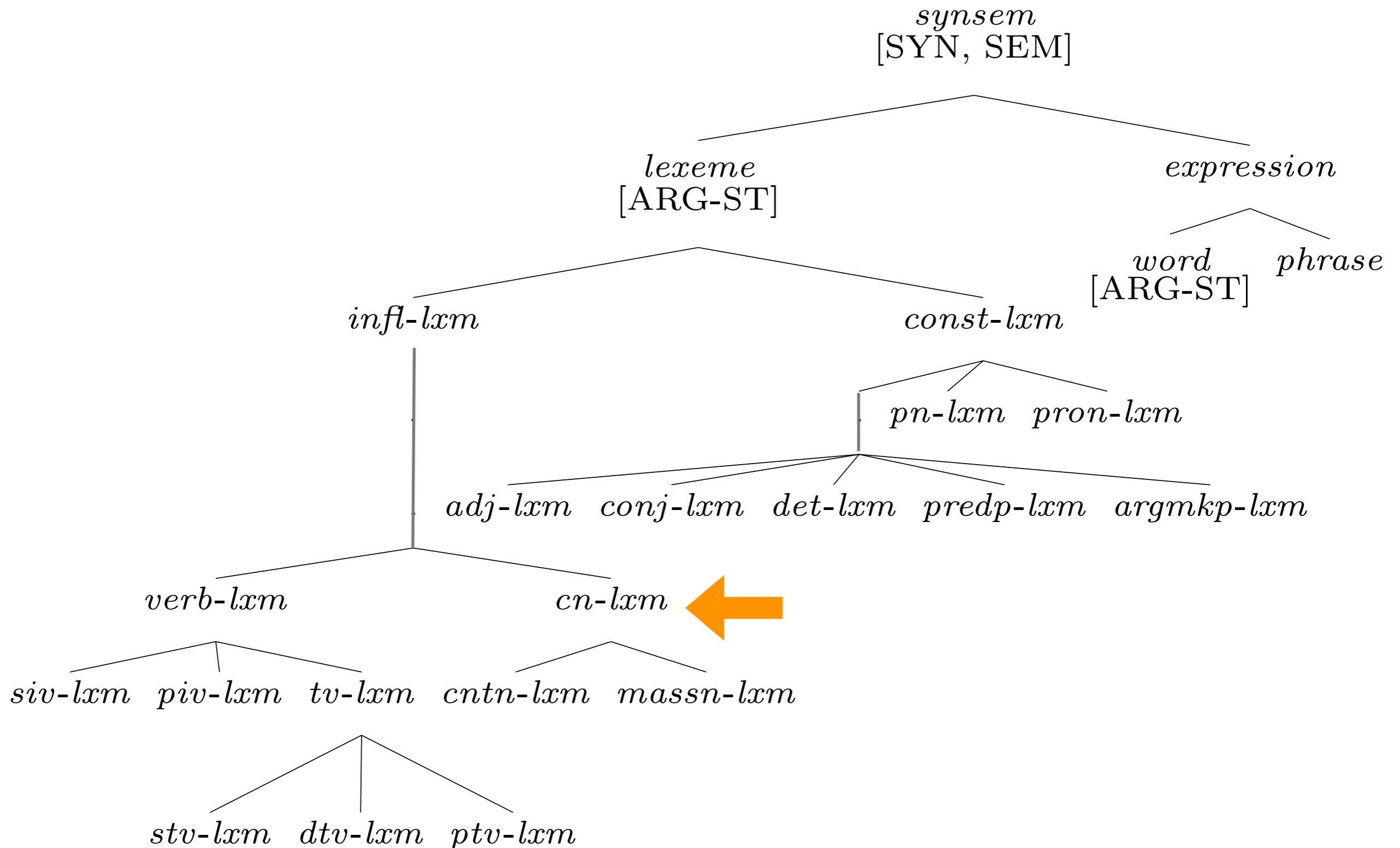
A Constraint on *infl-lxm*: the SHAC



A Constraint on *infl-lxm*: the SHAC

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

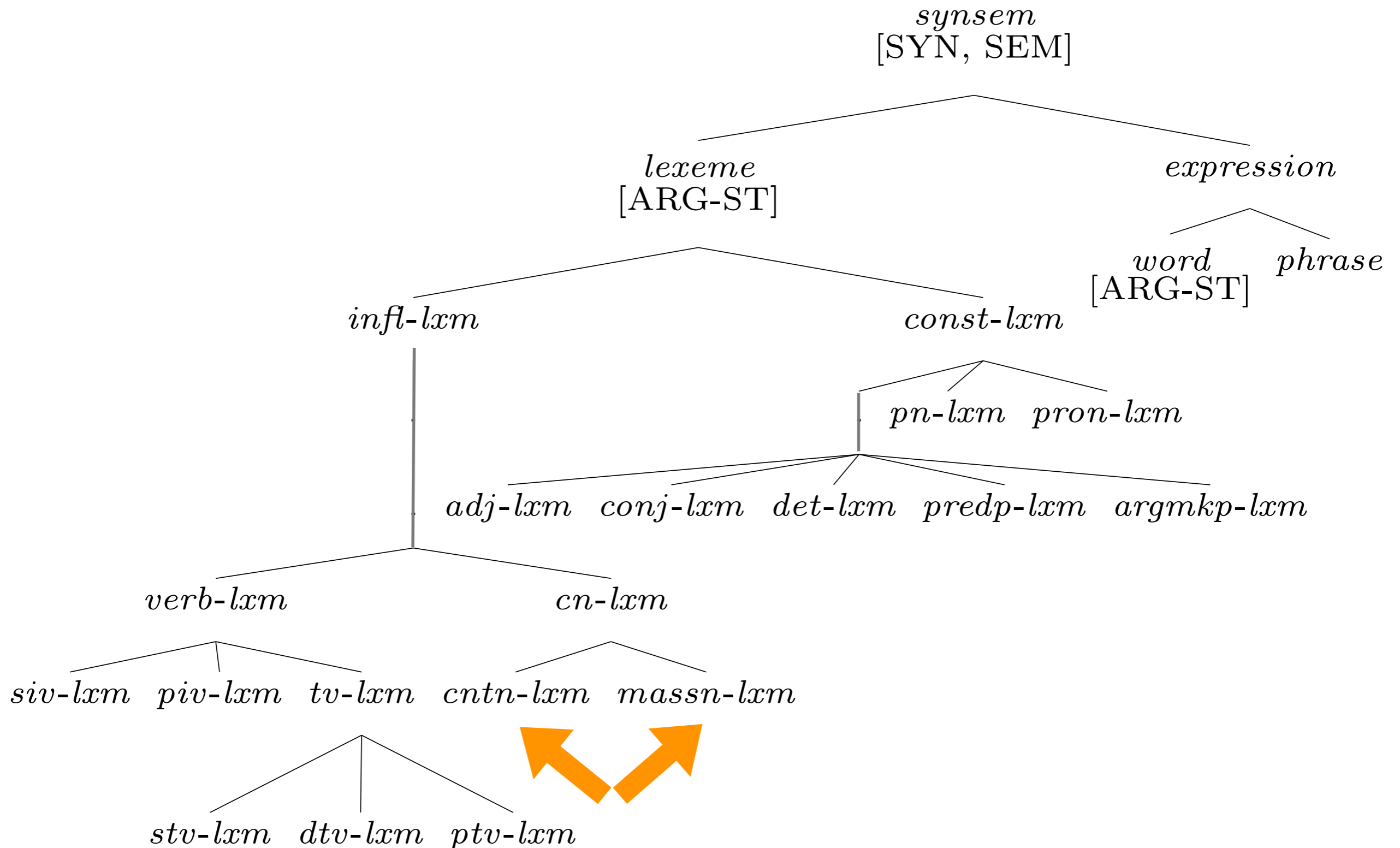
Constraints on *cn-lxm*



Constraints on *cn-lxm*

$$\begin{array}{l}
 \text{SYN} \\
 \text{SEM} \\
 \text{ARG-ST}
 \end{array}
 \left[
 \begin{array}{l}
 \left[
 \begin{array}{l}
 \text{HEAD} \\
 \text{VAL} \\
 \text{MODE} \\
 \text{INDEX}
 \end{array}
 \right]
 \left[
 \begin{array}{l}
 \left[
 \begin{array}{l}
 \text{noun} \\
 \text{AGR} \text{ [PER 3rd]}
 \end{array}
 \right] \\
 \left[
 \begin{array}{l}
 \text{SPR} \left\langle \left[\begin{array}{l} \text{HEAD} \\ \text{INDEX} \end{array} \right] \text{det} \right\rangle \\
 / \text{ref} \\
 i
 \end{array}
 \right] \\
 \langle X \rangle \oplus // \langle \rangle
 \end{array}
 \right]
 \end{array}
 \right]$$

Formally Distinguishing Count vs. Mass Nouns

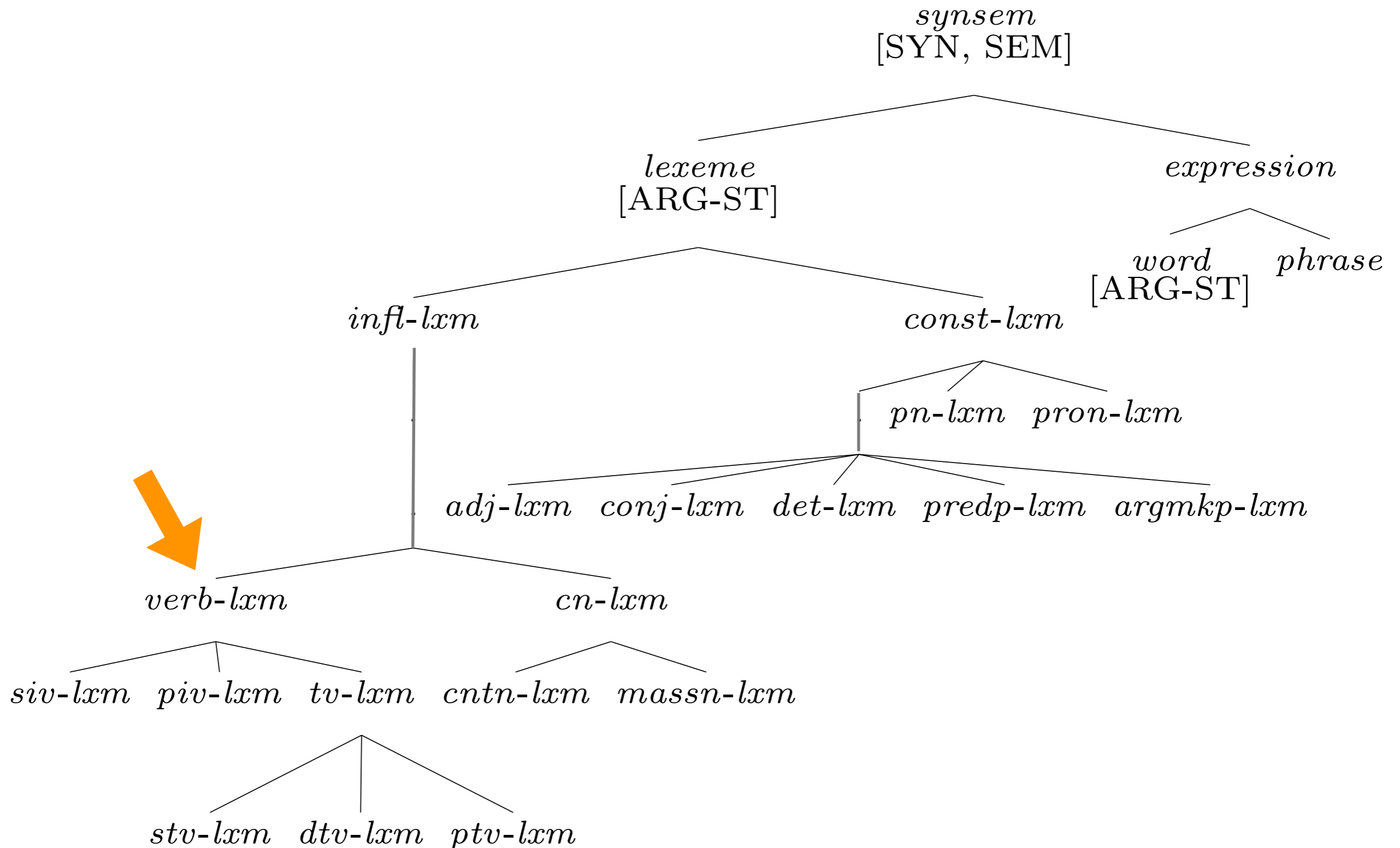


Formally Distinguishing Count vs. Mass Nouns

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

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

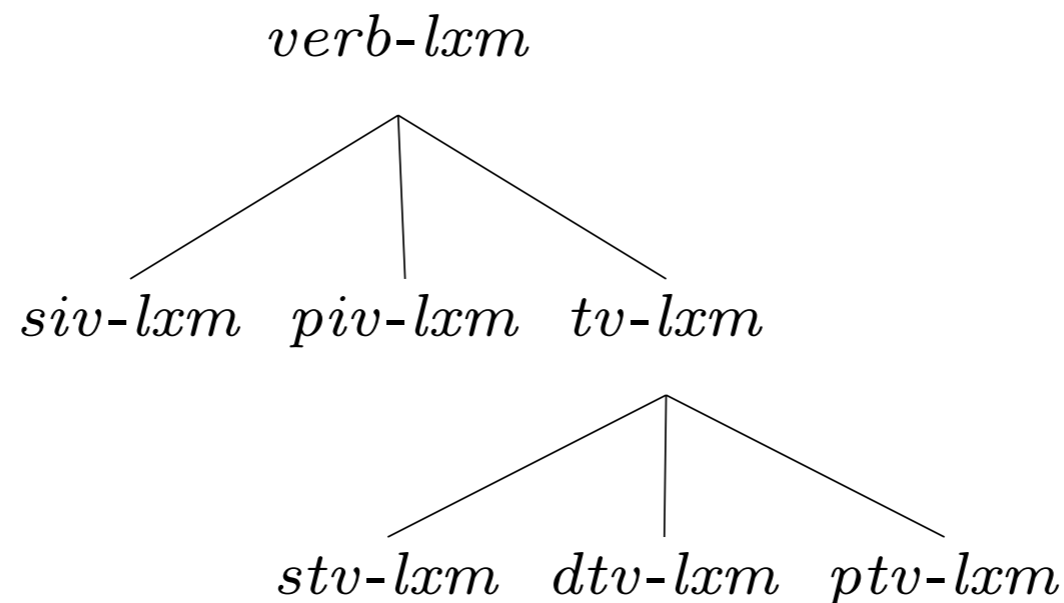
Constraints on *verb-lxm*



Constraints on *verb-lxm*

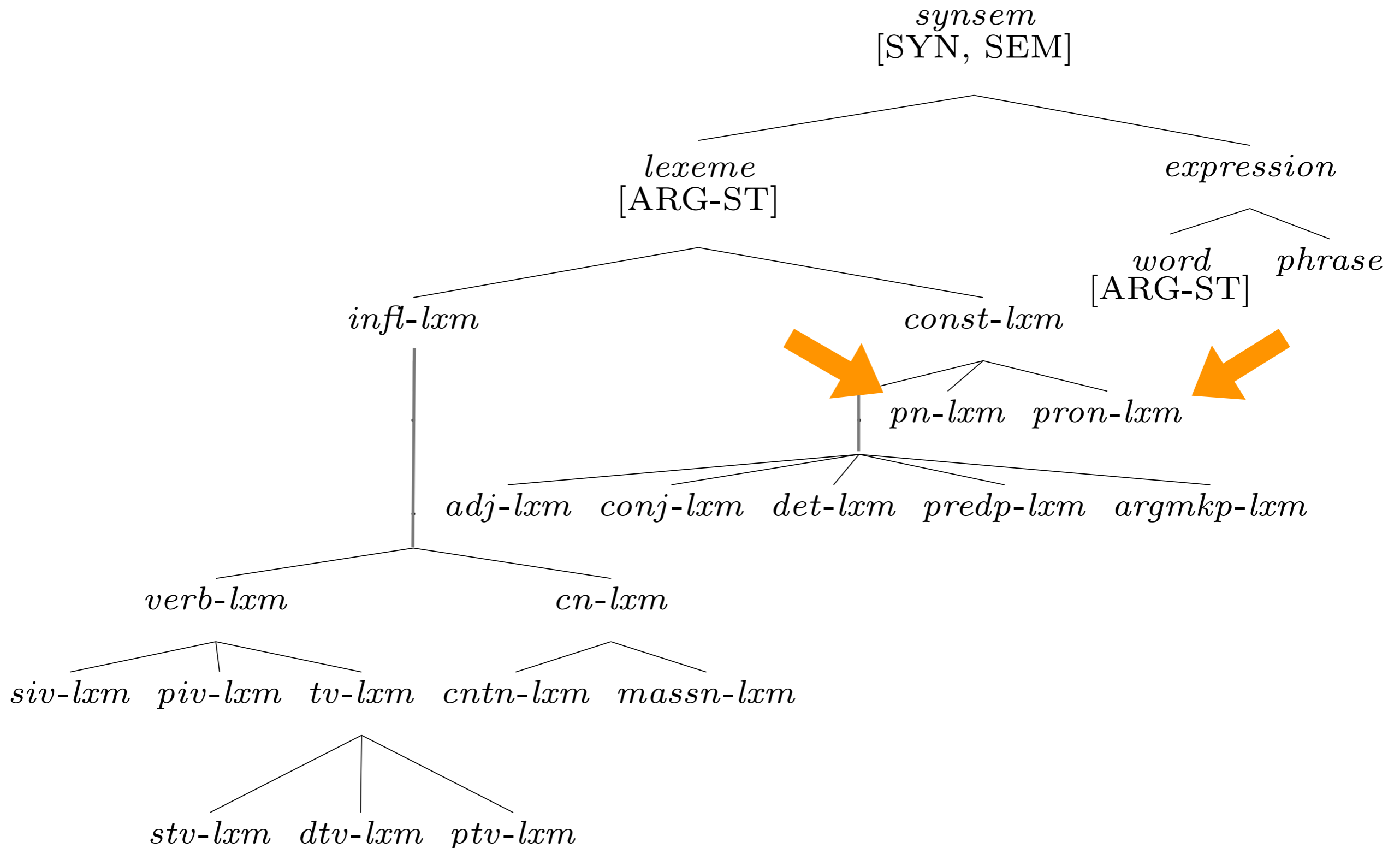
verb-lxm:
$$\left[\begin{array}{l} \text{SYN} \\ \text{SEM} \\ \text{ARG-ST} \end{array} \left[\begin{array}{l} \left[\text{HEAD} \quad \textit{verb} \right] \\ \left[\text{MODE} \quad \textit{prop} \right] \\ / \langle \text{NP}, \dots \rangle \end{array} \right] \right]$$

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:

$$\left[\begin{array}{l} \text{SYN} \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \textit{noun} \\ \text{AGR} \left[\begin{array}{l} \text{PER} \quad 3\text{rd} \\ \text{NUM} \quad / \text{sg} \end{array} \right] \end{array} \right] \end{array} \right] \\ \text{SEM} \left[\begin{array}{l} \text{MODE} \quad \text{ref} \end{array} \right] \\ \text{ARG-ST} \quad / \langle \rangle \end{array} \right]$$

pron-lxm:

$$\left[\begin{array}{l} \text{SYN} \left[\begin{array}{l} \text{HEAD} \quad \textit{noun} \end{array} \right] \\ \text{SEM} \left[\begin{array}{l} \text{MODE} \quad / \text{ref} \end{array} \right] \\ \text{ARG-ST} \quad \langle \rangle \end{array} \right]$$

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 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*