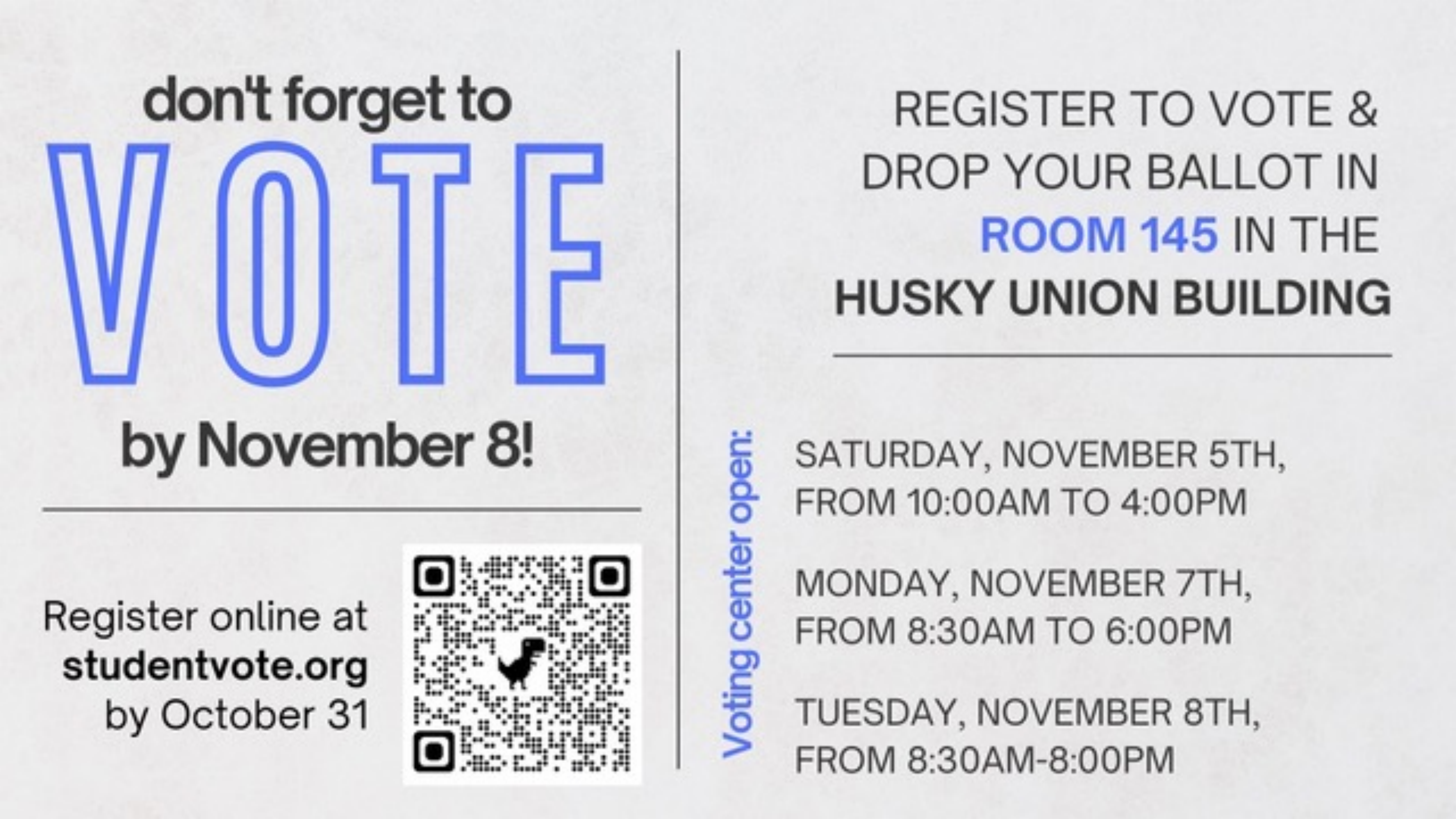
Ling 566
Oct 27, 2022

Lexical Types



don't forget to

VOTE

by November 8!

Register online at
studentvote.org
by October 31

REGISTER TO VOTE &
DROP YOUR BALLOT IN
ROOM 145 IN THE
HUSKY UNION BUILDING

Voting center open:

SATURDAY, NOVEMBER 5TH,
FROM 10:00AM TO 4:00PM

MONDAY, NOVEMBER 7TH,
FROM 8:30AM TO 6:00PM

TUESDAY, NOVEMBER 8TH,
FROM 8:30AM-8:00PM

Overview

- Motivation for lexical hierarchy
- Default inheritance
- Tour of the lexeme hierarchy
- The Case Constraint
- *pos* vs. *lexeme*
- Reading Questions

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

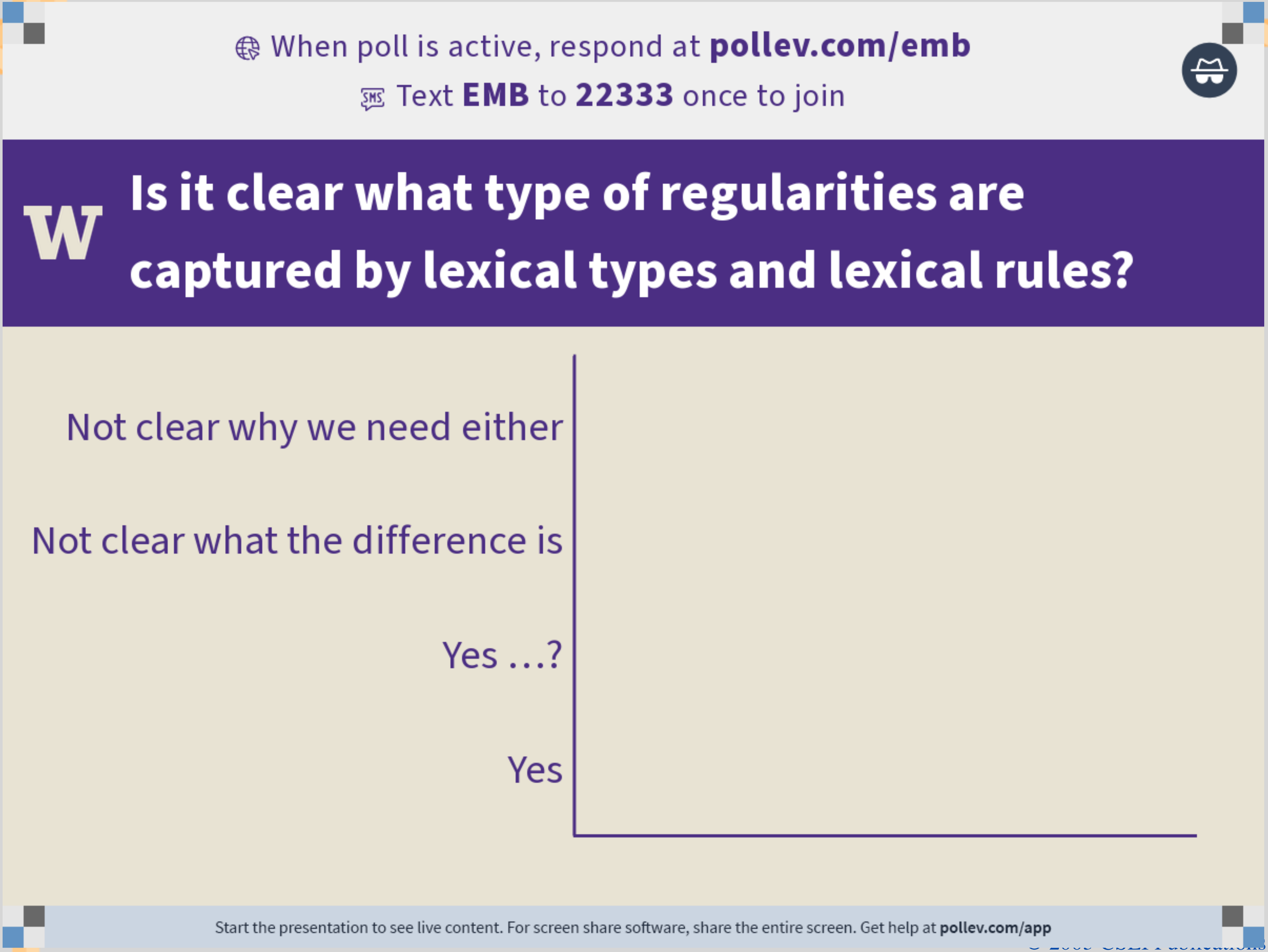
Q: Is lexeme the same as lemma?

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*,...



When poll is active, respond at **pollev.com/emb**

Text **EMB** to **22333** once to join

Is it clear what type of regularities are captured by lexical types and lexical rules?

Not clear why we need either

Not clear what the difference is

Yes …?

Yes

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

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 MOD / < S >, and one of its subtypes says MOD <[SPR < NP>] >, then the MOD value of instances of the subtype is what?

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

- That is, default constraints are ‘pushed down’

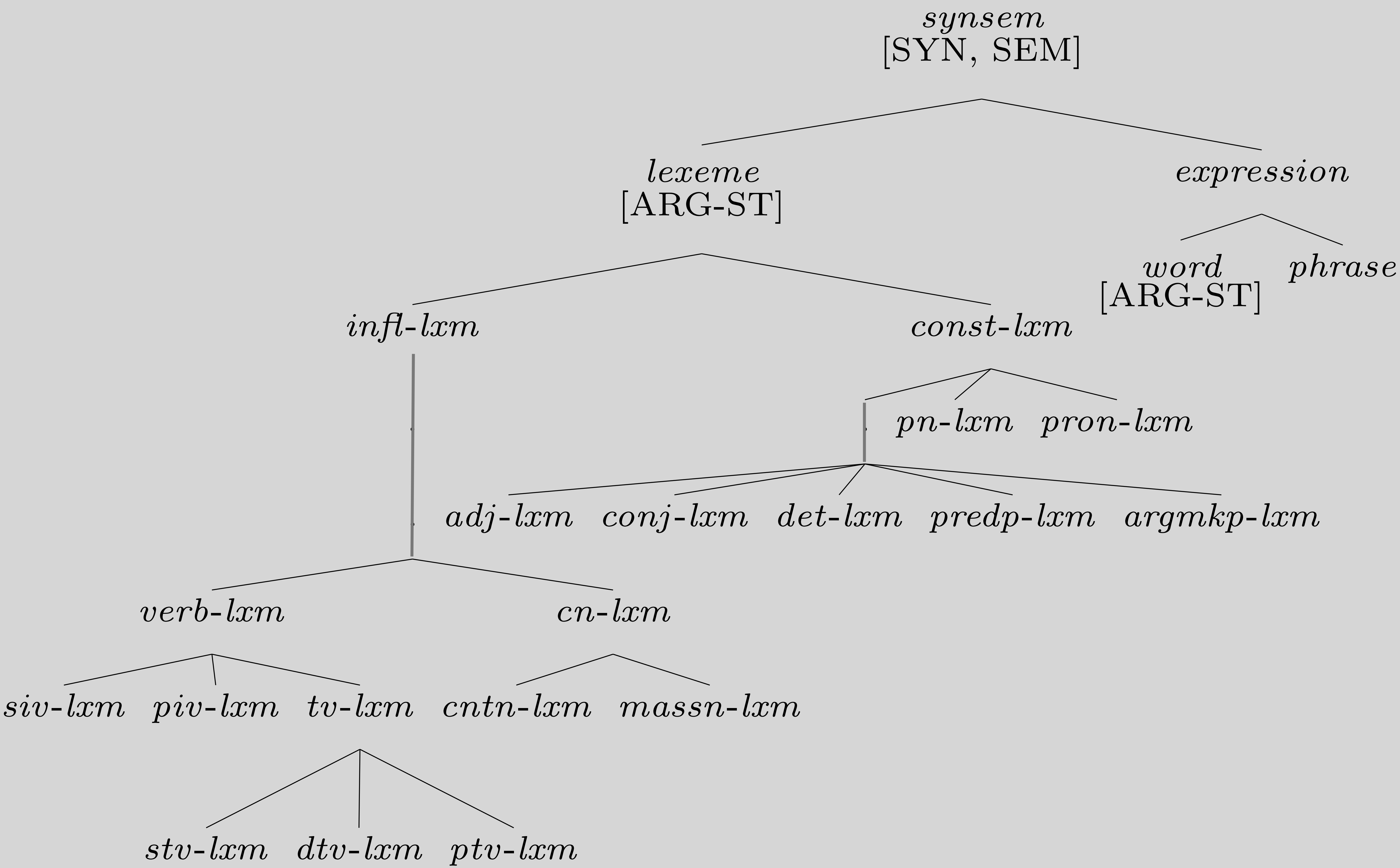
Question on Default Inheritance

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
[ARG-ST]

phrase

infl-lxm

const-lxm

pn-lxm *pron-lxm*

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

verb-lxm

cn-lxm

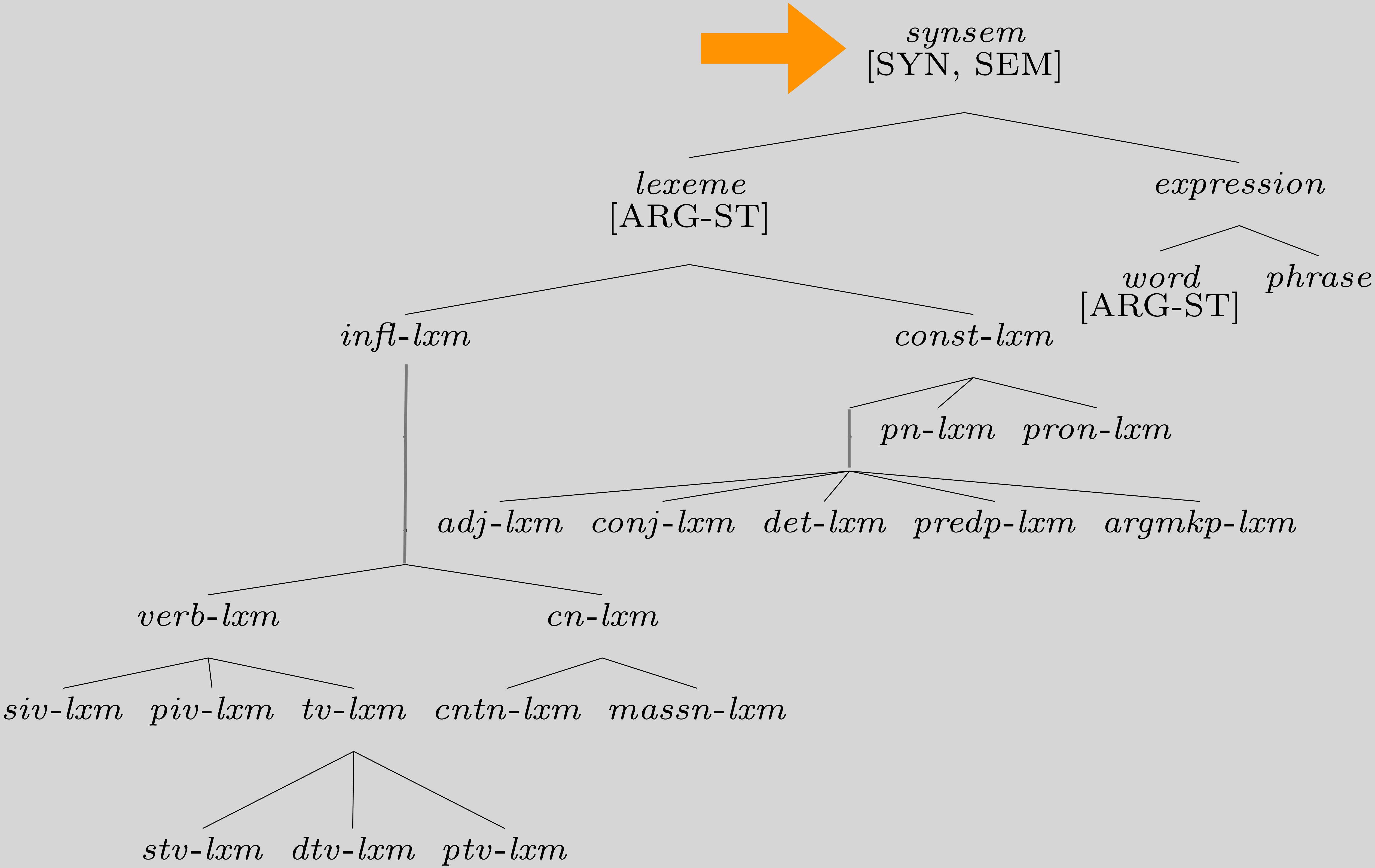
siv-lxm *piv-lxm* *tv-lxm* *cntn-lxm* *massn-lxm*

stv-lxm *dtv-lxm* *ptv-lxm*

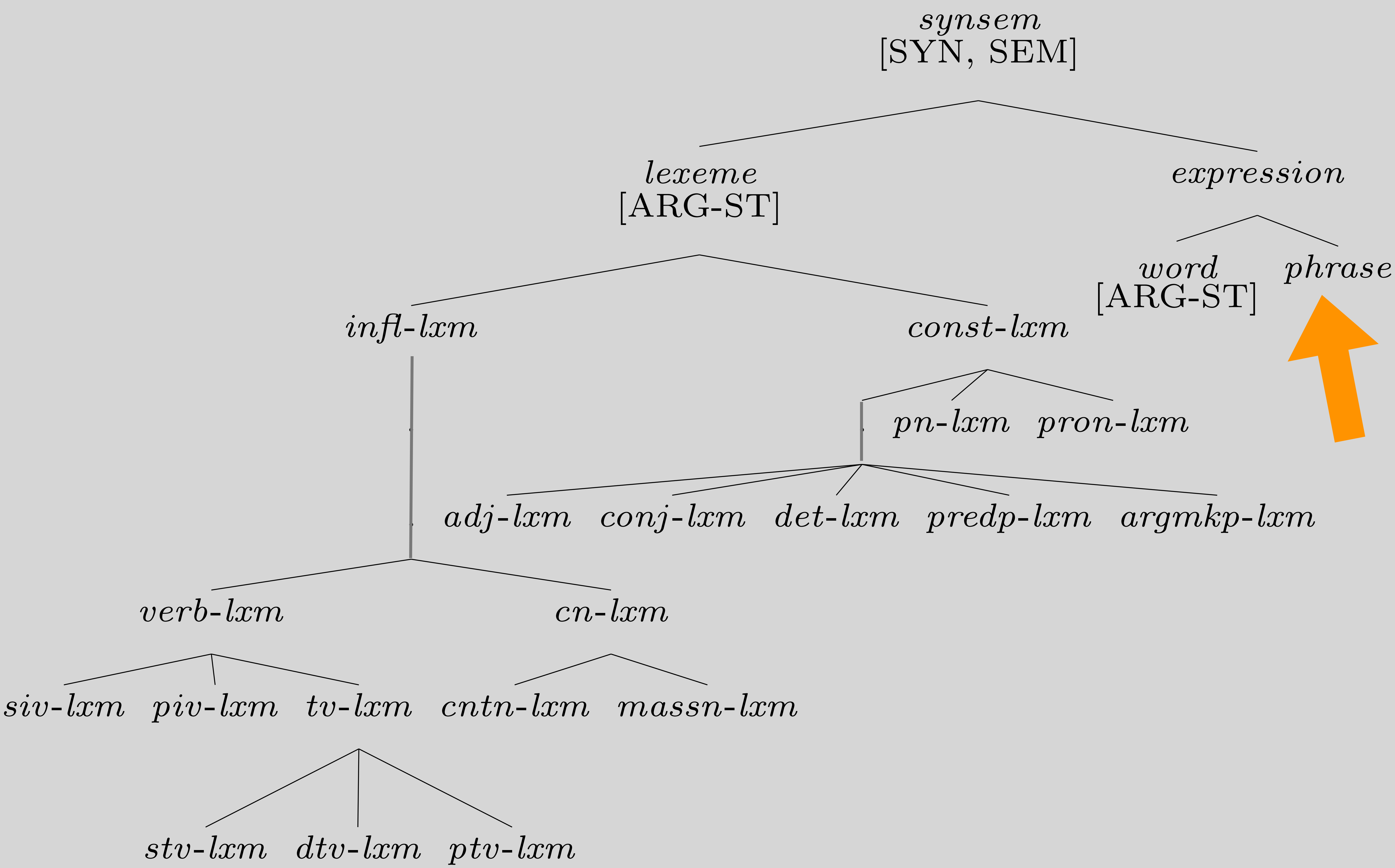
Functions of Types

- 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

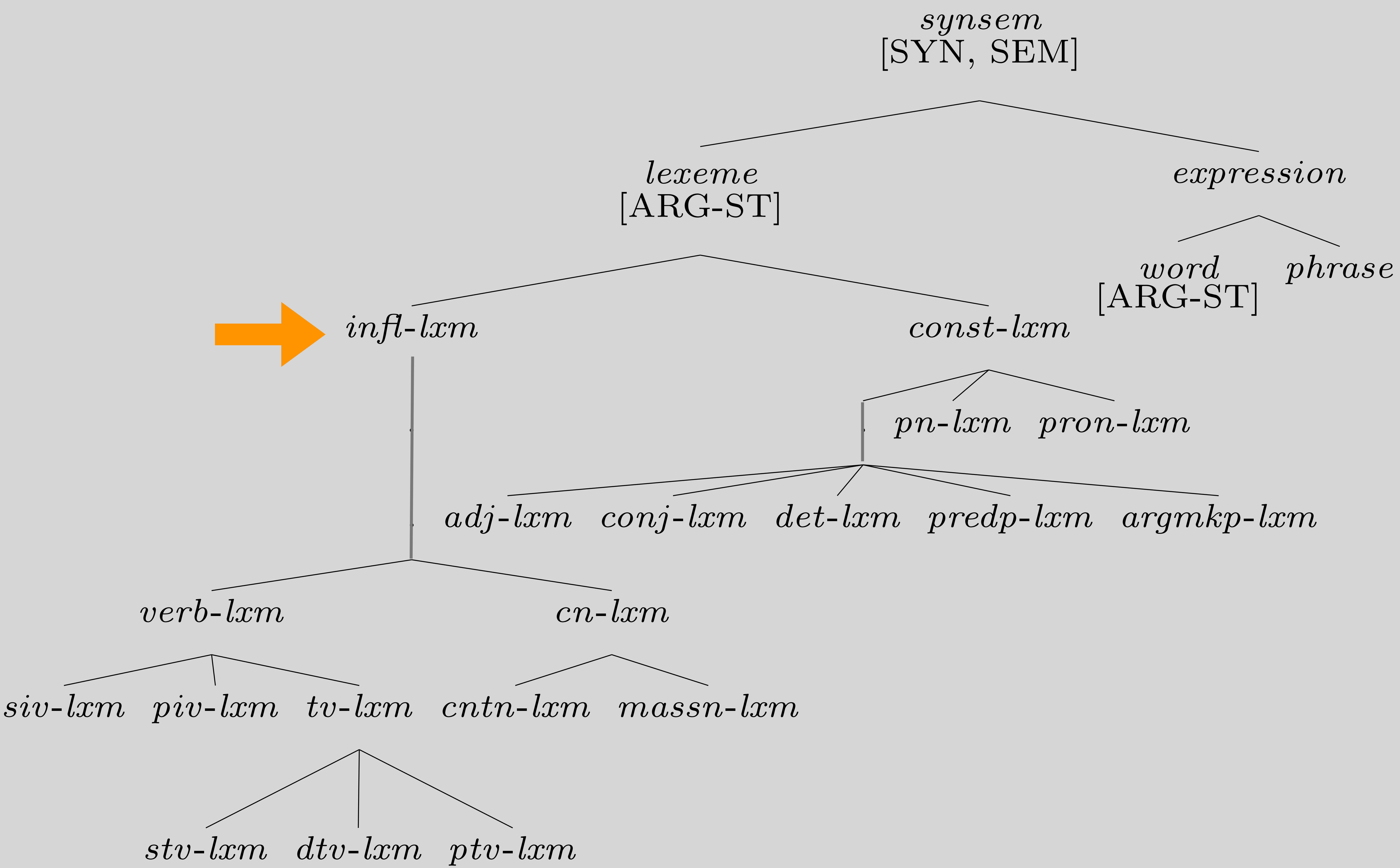
Every *synsem* has the features SYN and SEM



No ARG-ST on *phrase*



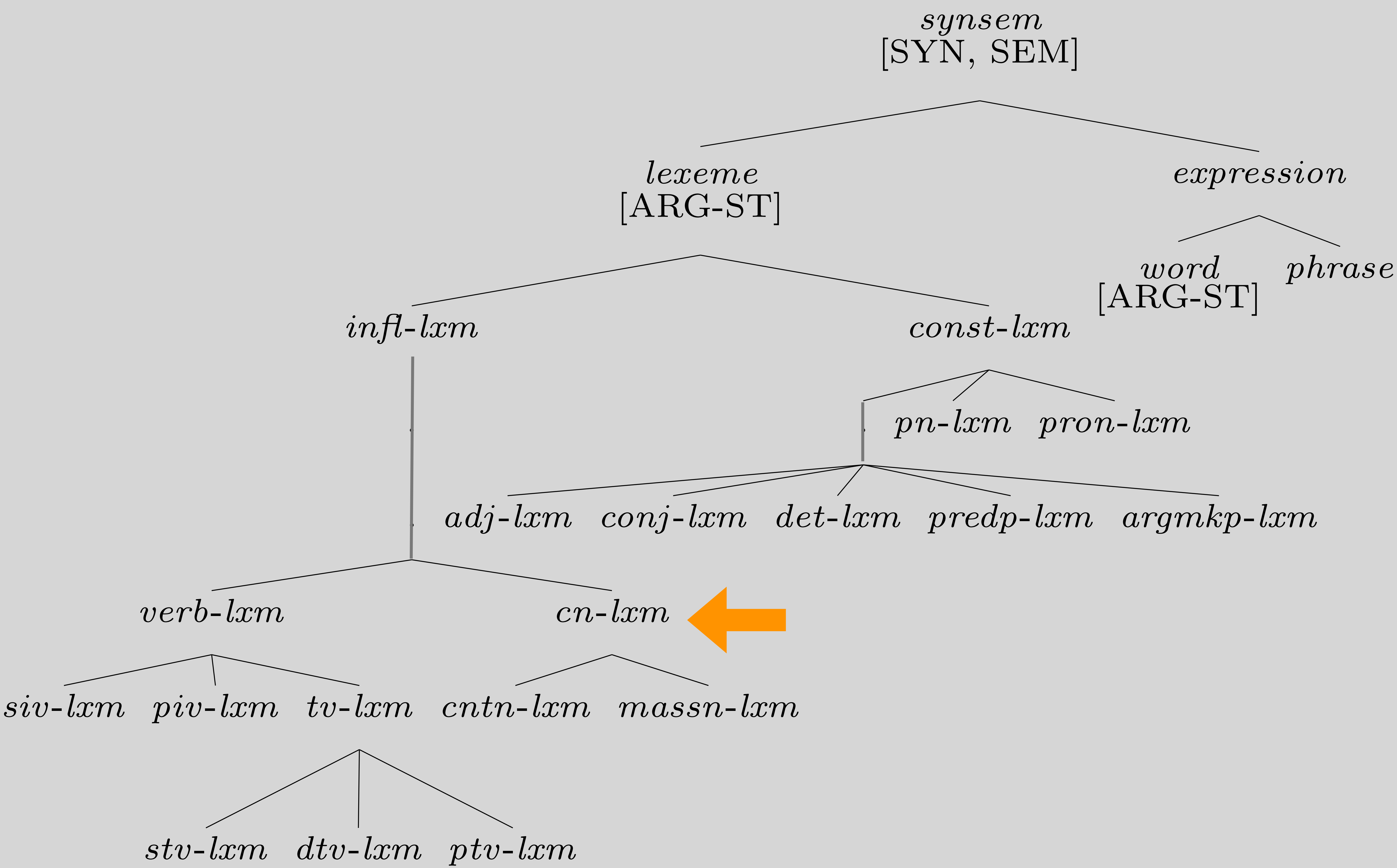
A Constraint on *infl-lxm*: the SHAC



A Constraint on *infl-lxm*: the SHAC

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

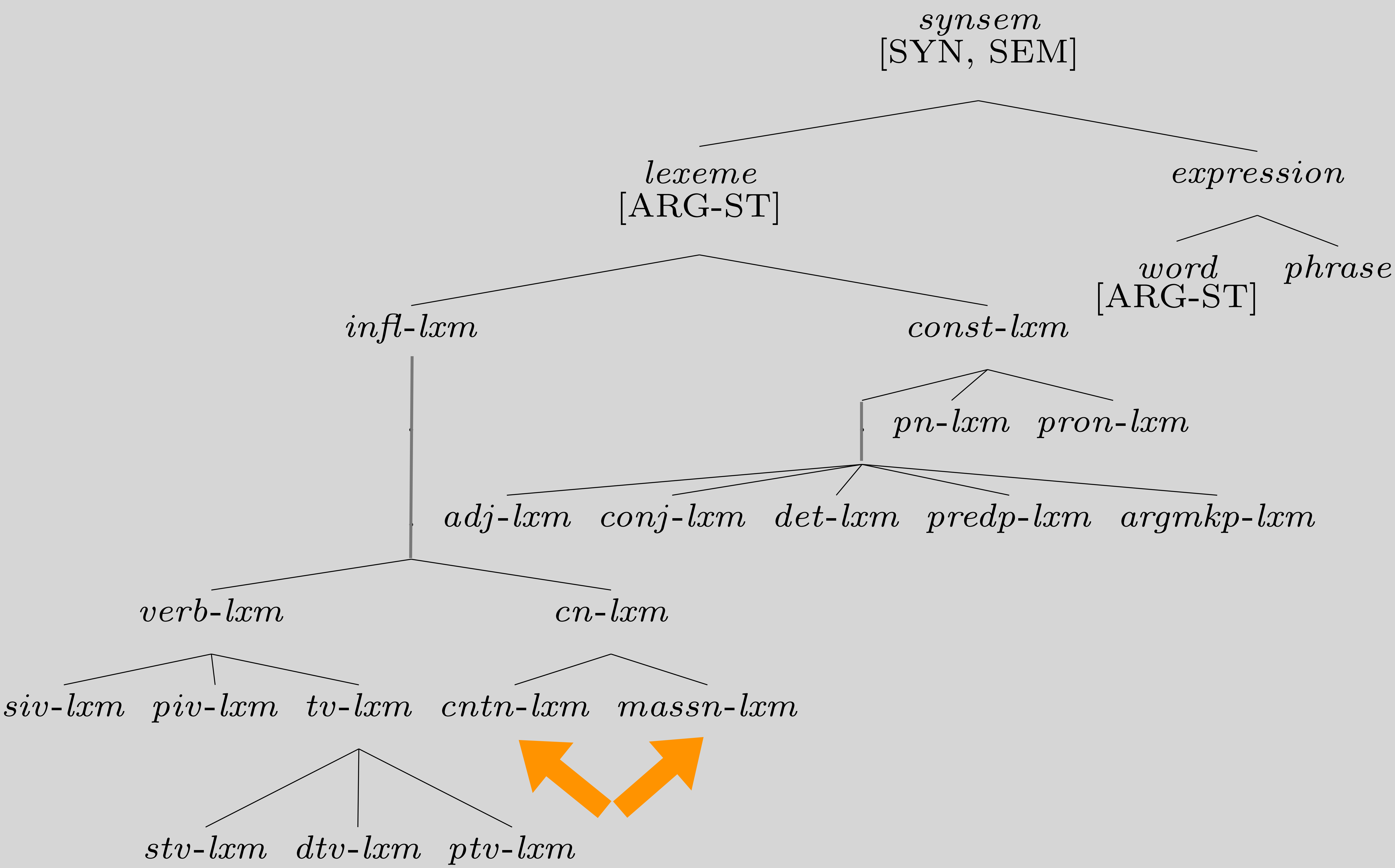
Constraints on *cn-lxm*



Constraints on *cn-lxm*

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

Formally Distinguishing Count vs. Mass Nouns

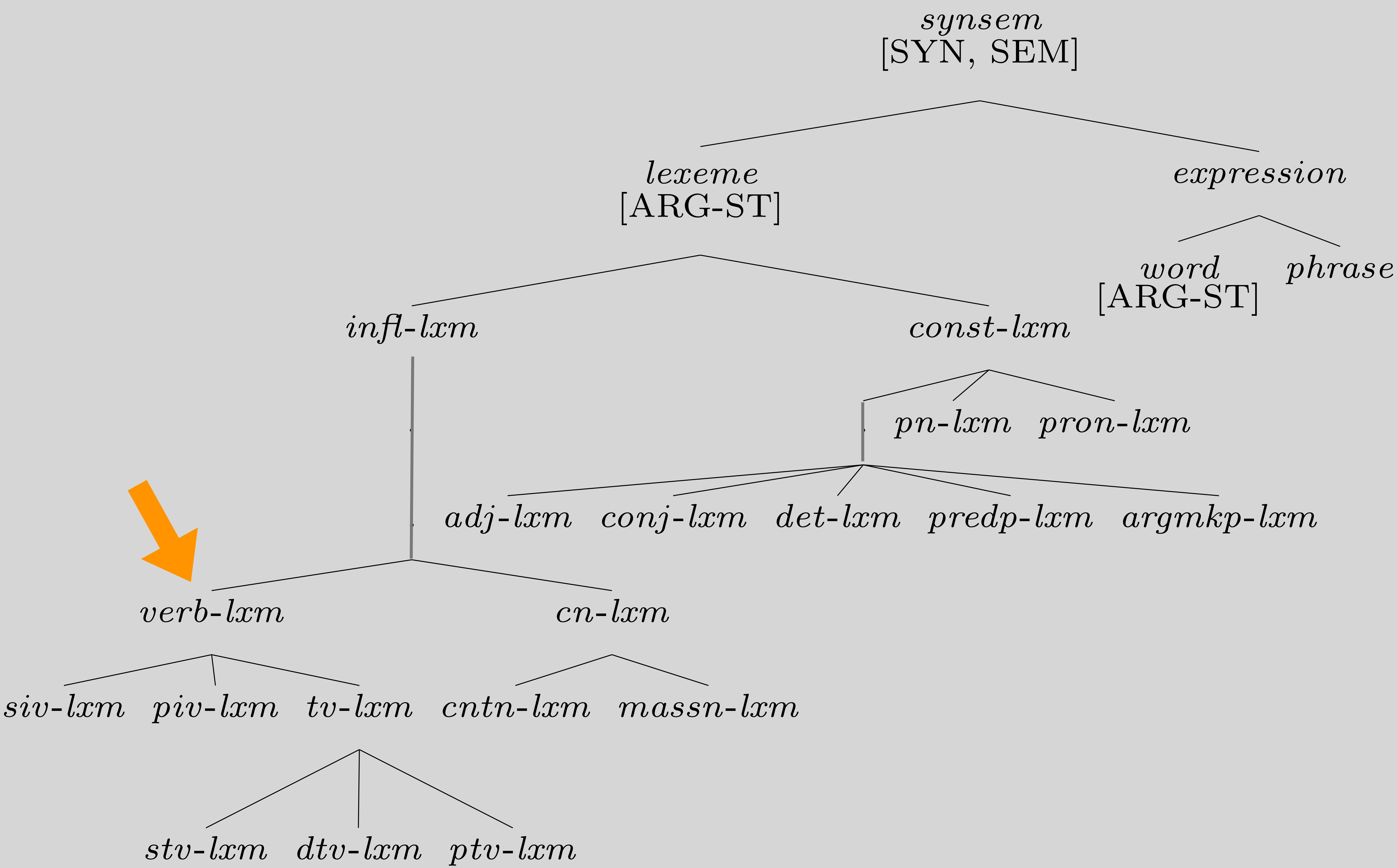


Formally Distinguishing Count vs. Mass Nouns

$$\textit{cntn-lxm} : \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ [\text{COUNT}\ +]\ \rangle\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

$$\textit{massn-lxm} : \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ [\text{COUNT}\ -]\ \rangle\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

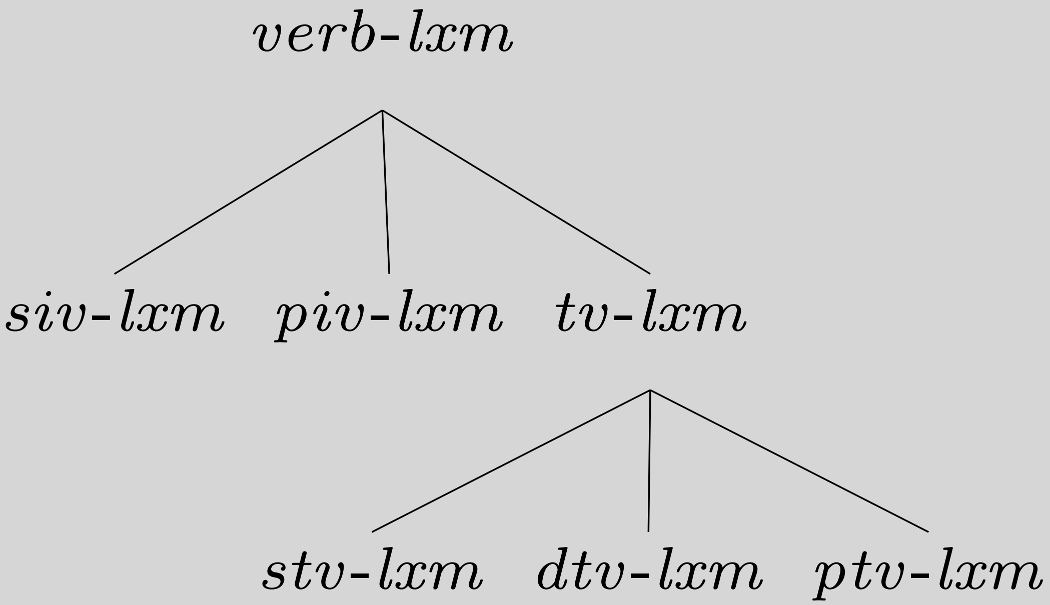
Constraints on *verb-lxm*



Constraints on *verb-lxm*

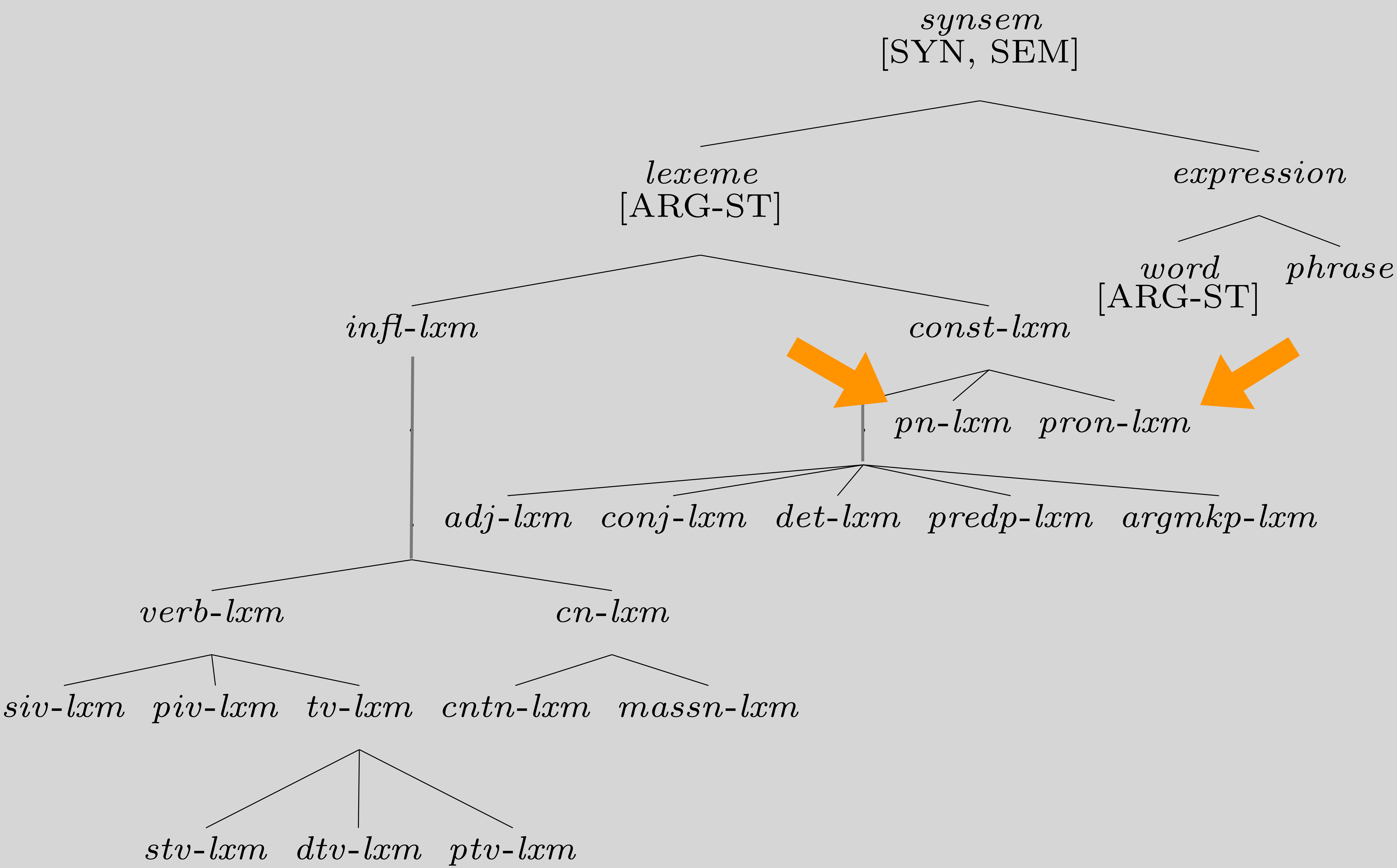
$$\textit{verb-lxm}: \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \end{bmatrix} \\ \text{ARG-ST} & / \left\langle \text{NP}, \ldots \right\rangle \end{bmatrix}$$

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

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

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

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?

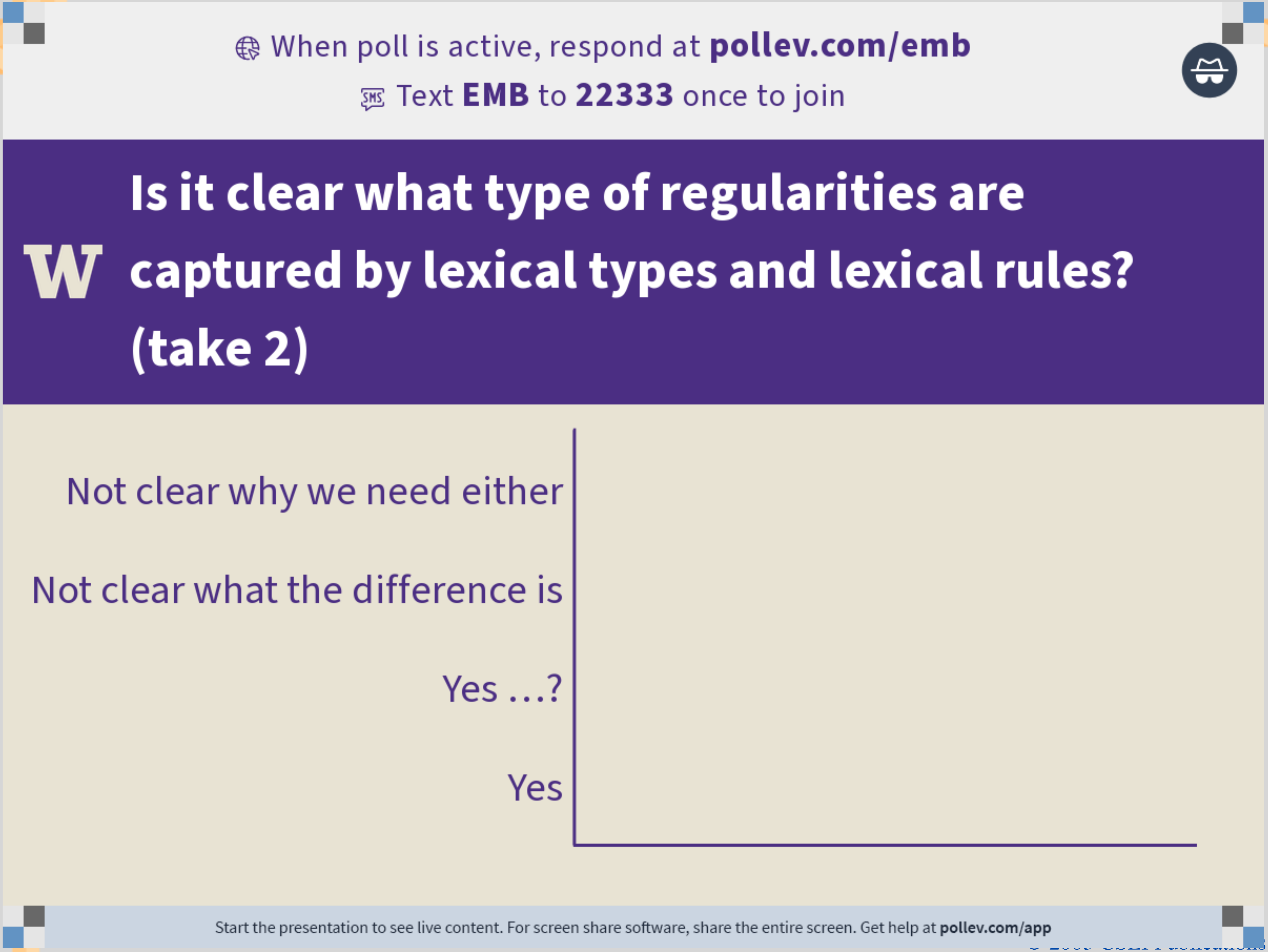
  No: The Case Constraint is an English-specific constraint.

Apparent redundancy

- Why do we need both the *pos* subhierarchy and lexeme types?
- *pos*:
  - Applies to words and phrases; models relationship between 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*,...



When poll is active, respond at **pollev.com/emb**

Text **EMB** to **22333** once to join

Is it clear what type of regularities are captured by lexical types and lexical rules? (take 2)

- Not clear why we need either
- Not clear what the difference is
- Yes …?
- Yes

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Overview

- Motivation for lexical hierarchy
- Default inheritance
- Tour of the lexeme hierarchy
- The Case Constraint
- *pos* vs. *lexeme*
- Reading Questions

HW4 tips

- Ch 7 Problem 1:
  - Not grading you on the judgments, but on the sentences constructed and matching classification to the judgments
  - Be sure to keep the same verb + preposition pair
- Ch 8 grammar summary is in Ch 9

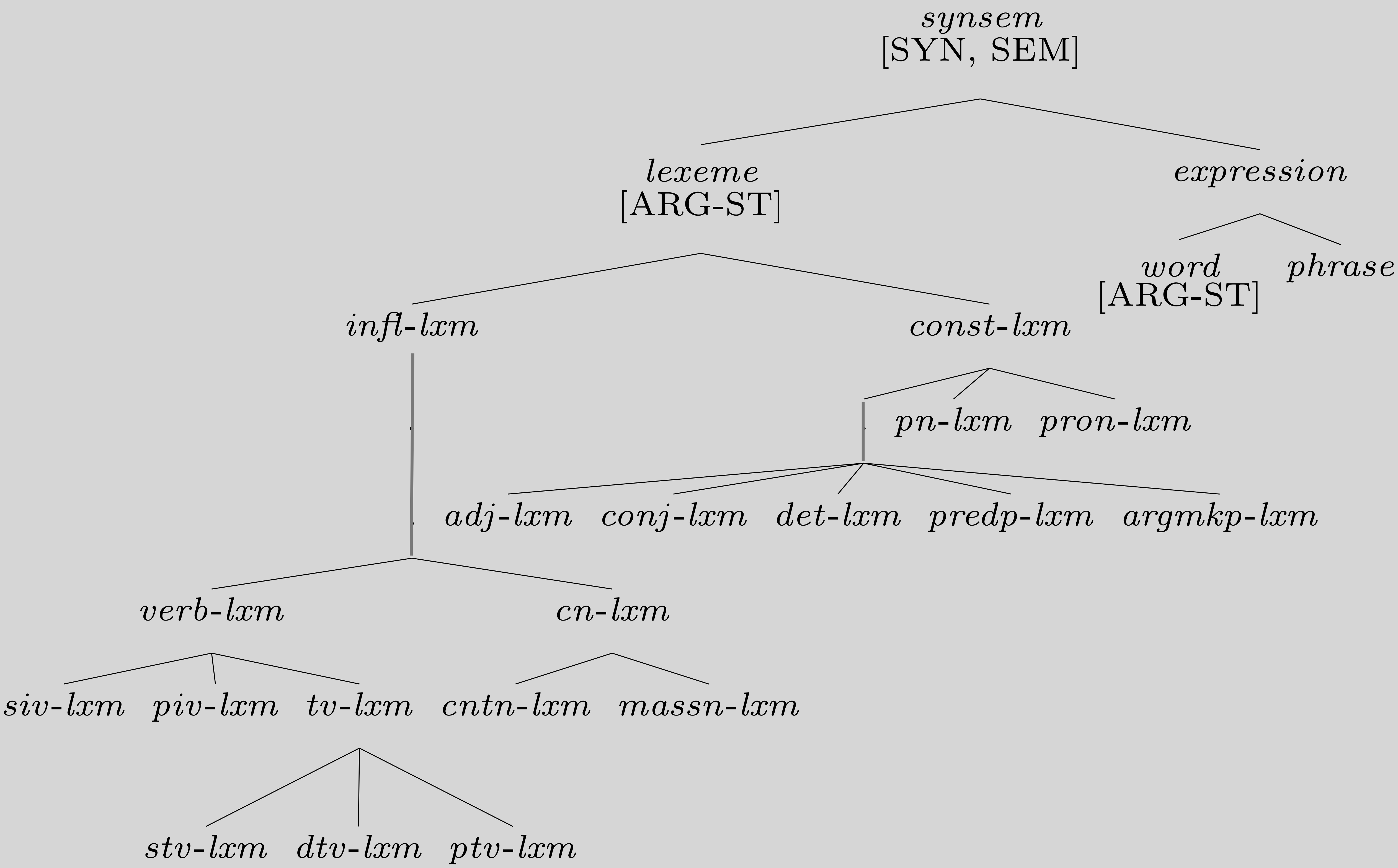
Reading Questions

- The letters X, Y, Z are used in the ARG-ST of several lexeme types. Is there significance to the use of these letters?
    - No, these are just placeholders meaning "some feature structure". We could also have written [ ].

Reading Questions

- We've discussed before that words with multiple meanings will have multiple lexical entries. Do we also have different lexemes?
- Why aren't *lexemes* *expressions* in the type hierarchy?

Our Lexeme Hierarchy



Reading Questions

- Why do we introduce *synsem* directly under *feat-struc*? It seems to be the only entry on that level of hierarchy, and I am not sure how to intuitively understand what level of abstraction that tier of hierarchy captures.

Reading Questions

- Why are we putting ditransitive verbs under transitive verbs in our lexeme hierarchy? I thought we would separate these as different categories since strict transitive verbs require one complement and ditransitive verbs (and prepositional transitive) require two complements.
- On page 241, I was wondering why there isn't a type called intransitive-verb-lexeme which has two children: *siv-lxm* and *piv-lxm*.

Reading Questions

- What is the difference between "words with spaces in them" and "compound nouns"? How does this relate to the shaping the argument of letting the proper nouns select for specifiers?

Reading Questions

- How do we know when a constraint should be defeasible (is it fair to say default(?)) versus underspecified? In other words, if different variations on a constraint are grammatical, why even have defeasible constraints that can be overridden?

- Echoing the question on the difference between defeasible constraints and underspecification. My confusion is generally about this new introduction to the grammar and constraint inheritance, etc.

Reading Questions

- When we say a constraint holds by default, does it mean this constraint comes from some background information that's unrelated syntax (for instance, entity: [TEL / 555-111-1234], the value of the telephone number is set by default)?
- Could syntactic constraints, for instance, the value of SPR, COMPS, MOD, hold by default in some situations?

Reading Questions

- If a specification has a /, it is defeasible. So if a specification doesn't have /, then it must not change? Are their circumstances where it can be added to?
- In (21), there are two feature values that have /[1] (box 1). The slash means it is defeasible, so does the [1] mean that if one changes then the other must change to the same value?

Reading Questions

- Also, is the defeasibility related to the monotonicity? Like the words (such as nobody, often) may determine the nature of the whole sentences with a larger extent. Does it mean these words (which have high monotonicity level) have a high level of defeasibility on their constraints?

Reading Questions

- There are some rules that have a clause to the effect of "unless stated otherwise...". In general, could virtually any rule be overridden if a given grammar calls for it? At the end of the day, we are only describing language, so theoretically any of the rules could be broken?

Reading Questions

- Just to clarify 8.4.4 Lexemes vs. Parts of Speech, is it essentially saying it's a coincidence pos and lexeme have similar subtypes? These similarities are somewhat predictable in English but are there some languages where the subtypes for pos and lexeme are very different?

Reading Questions

- What's the difference between lexical entry & lexical sequence?
- What is a family of lexical sequences?

Reading Questions

- How does HPSG compare with other grammatical systems that are less information-heavy in the lexicon and what are some examples of grammar building systems that are more "information light" in the lexicon?
- Are there any particular languages for which information-heavy lexicons are particularly well or poorly suited?

Reading Questions

- The text explains the difference between the two types pos and lexeme, stating that the former specifies features that are appropriate for words and phrases whereas the latter constrains how such features can combine. This made me think about agglutinative languages, which encode a lot of grammatical information in morphology. Is it correct to assume that grammars for such languages would have a dense and diverse lexeme type hierarchy? Or is a different type required to model those languages well?

Reading Questions

- Now we are putting more and more weight on the lexical entries. I am wondering how this model would work or not work for languages with hazy word boundaris or even lexeme boundaries like Chinese.

Reading Questions

- Was this idea borne from OOP, or was this developed independently from OOP?
- Also, is this how much of the lexical items in HPSG are implemented in a computer, are lexical items implemented as objects that inherit specific values from their respective super types?