

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 (*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 is a formal mechanism for expressing such soft generalizations.

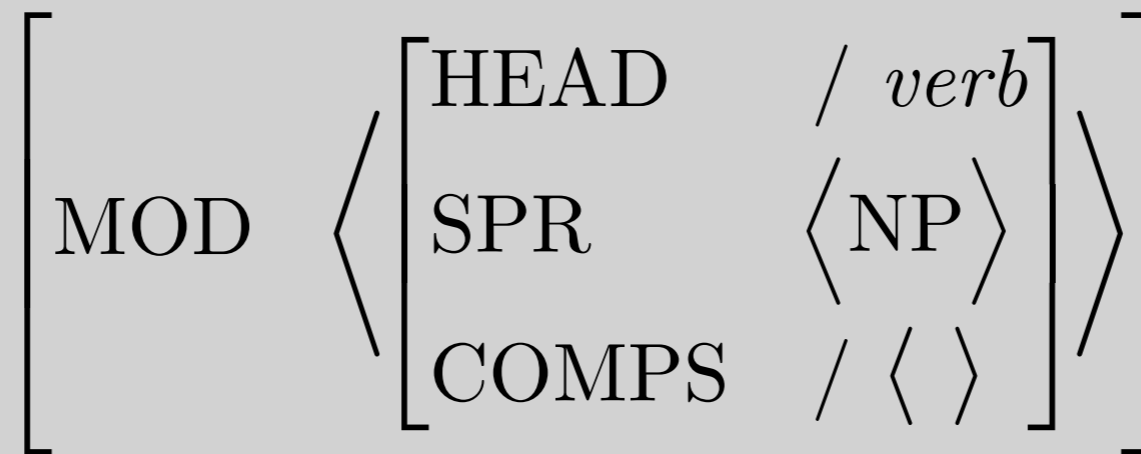
Default Inheritance, Technicalities

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 $\text{MOD} / \langle S \rangle$,
and one of its subtypes says $\text{MOD} \langle [\text{SPR} \langle \text{NP} \rangle] \rangle$,
then instances of the subtype are:

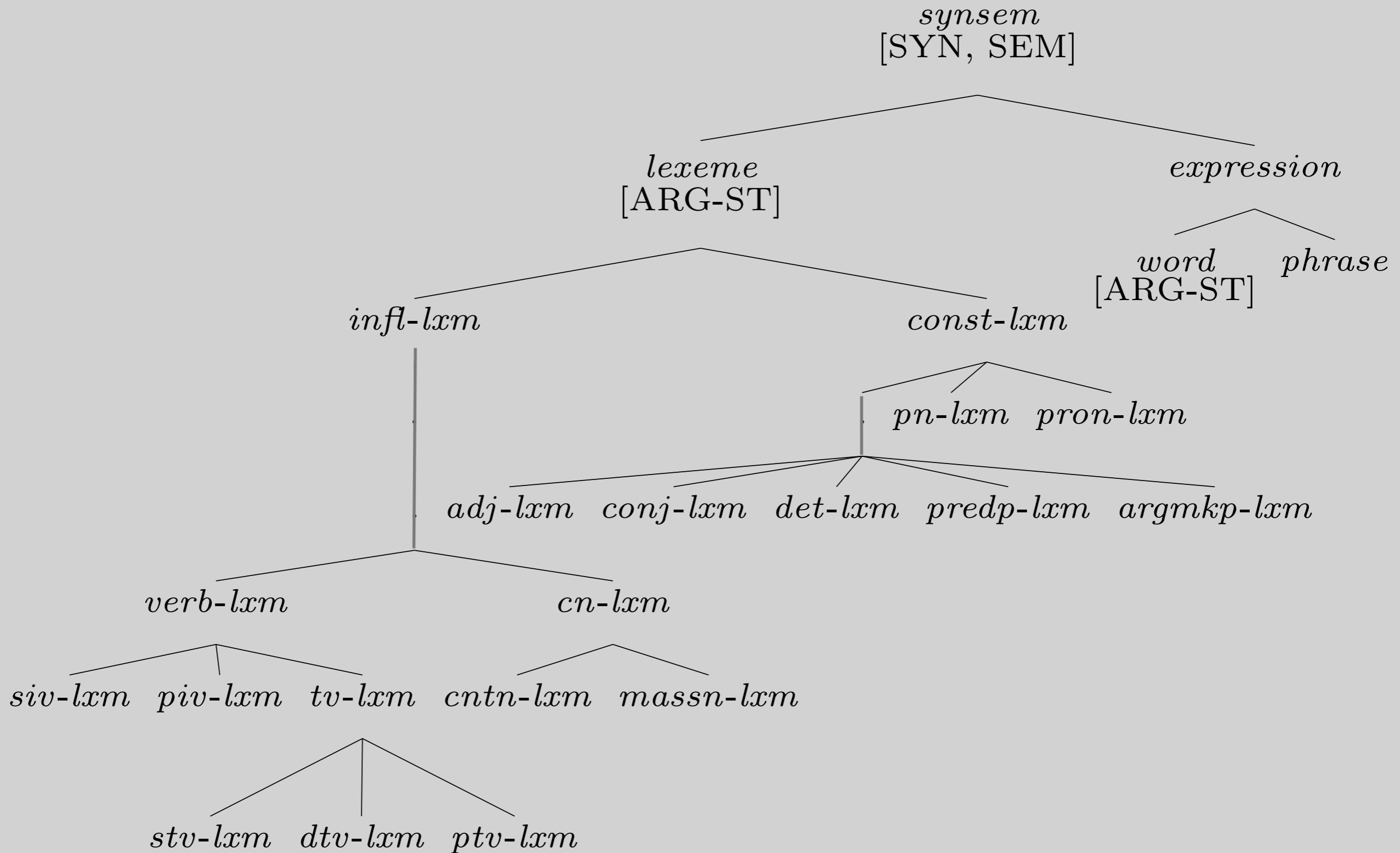


- That is, default constraints are ‘pushed down’

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.

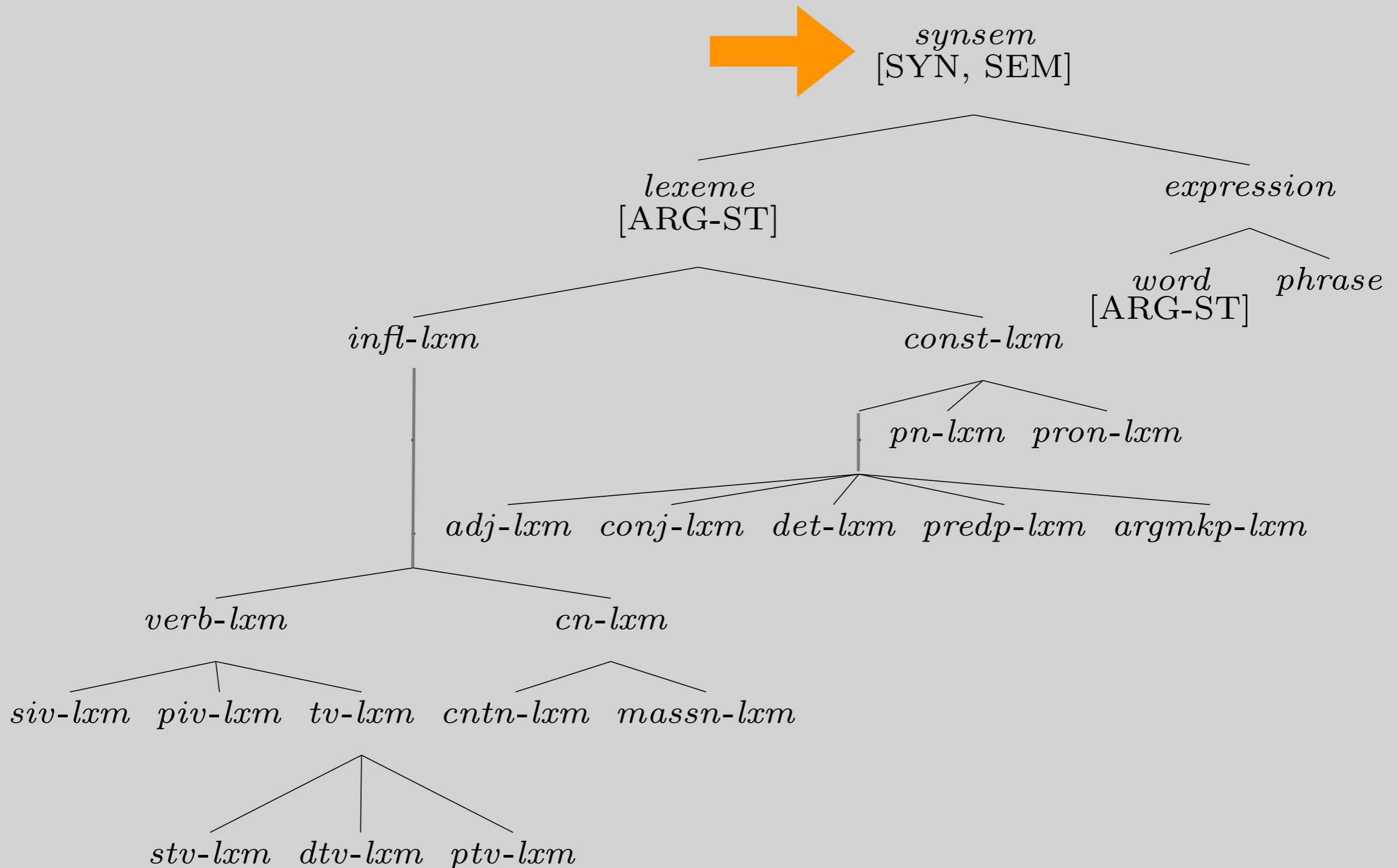
Our Lexeme Hierarchy



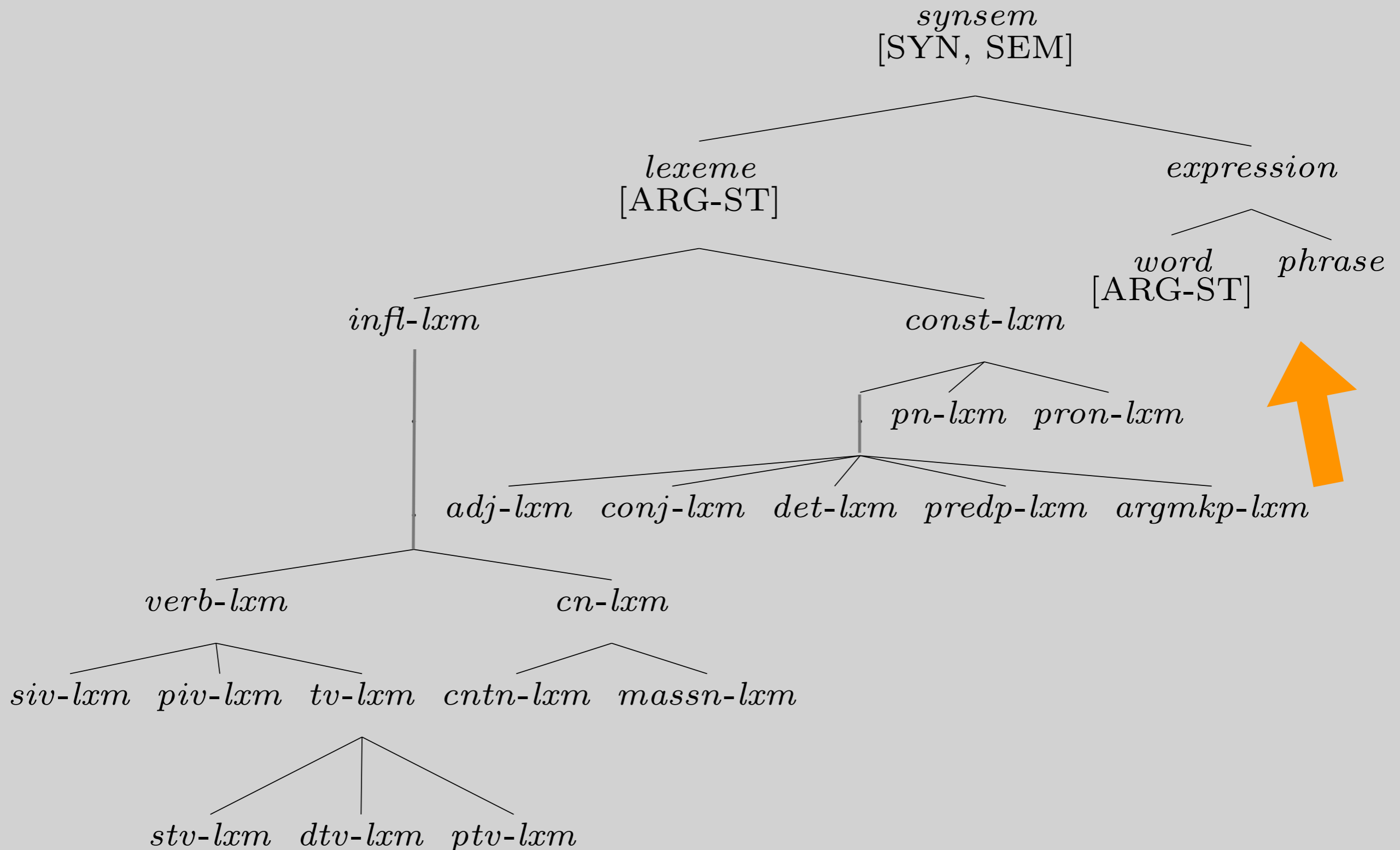
Functions of Types

- 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

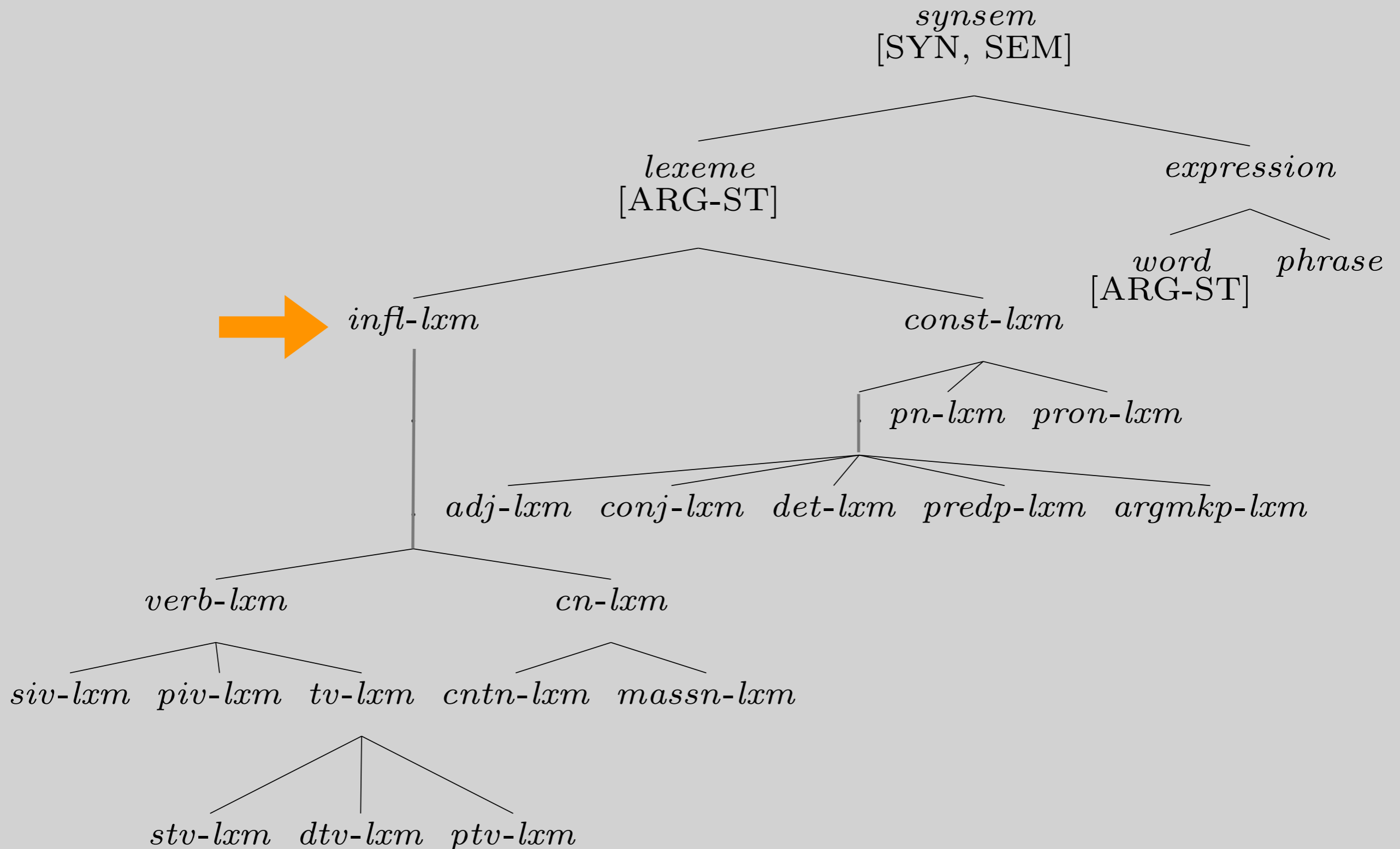
Every *synsem* has the features SYN and SEM



No ARG-ST on *phrase*



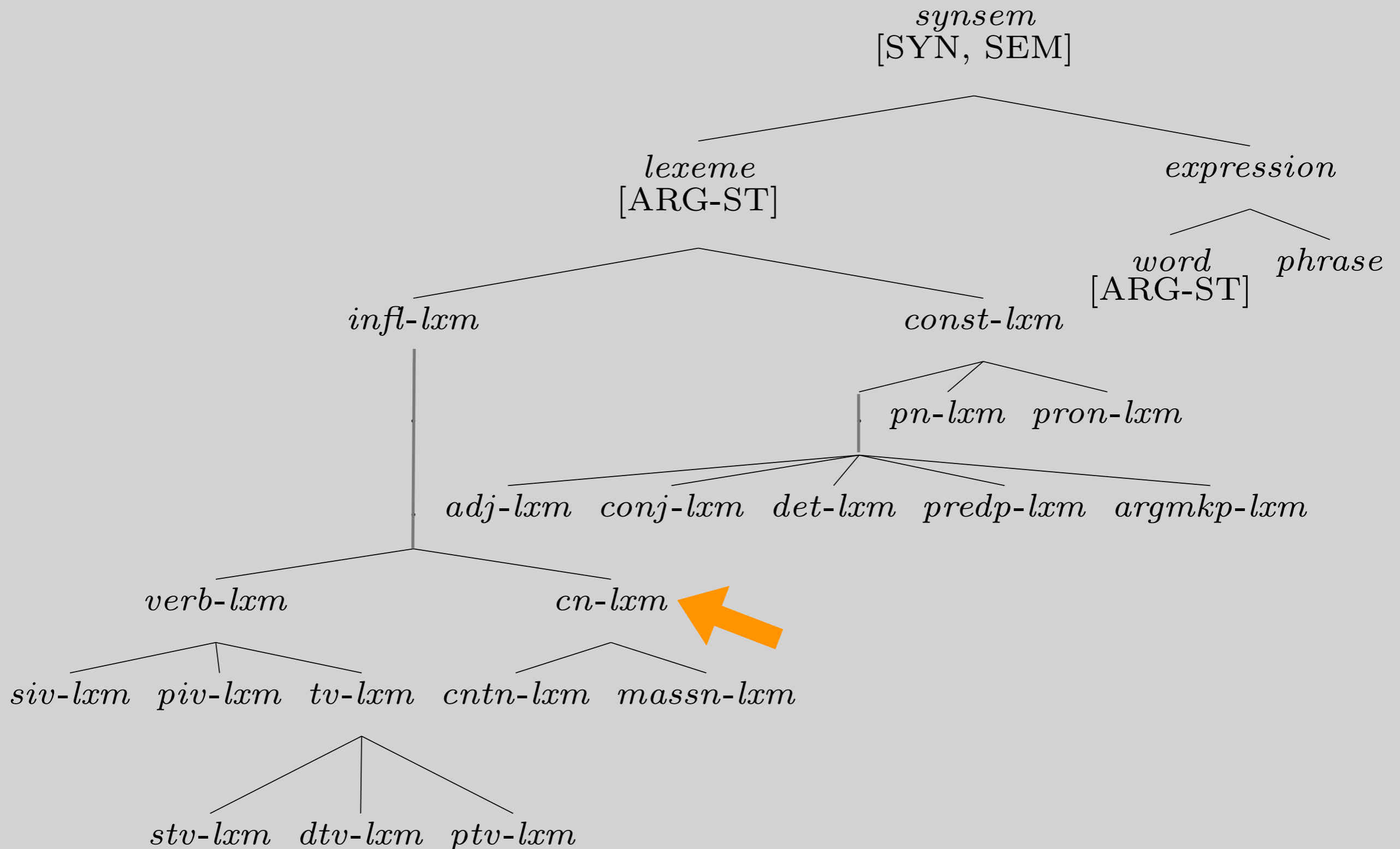
A Constraint on *infl-lxm*: the SHAC



A Constraint on *infl-lxm*: the SHAC

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

Constraints on *cn-lxm*

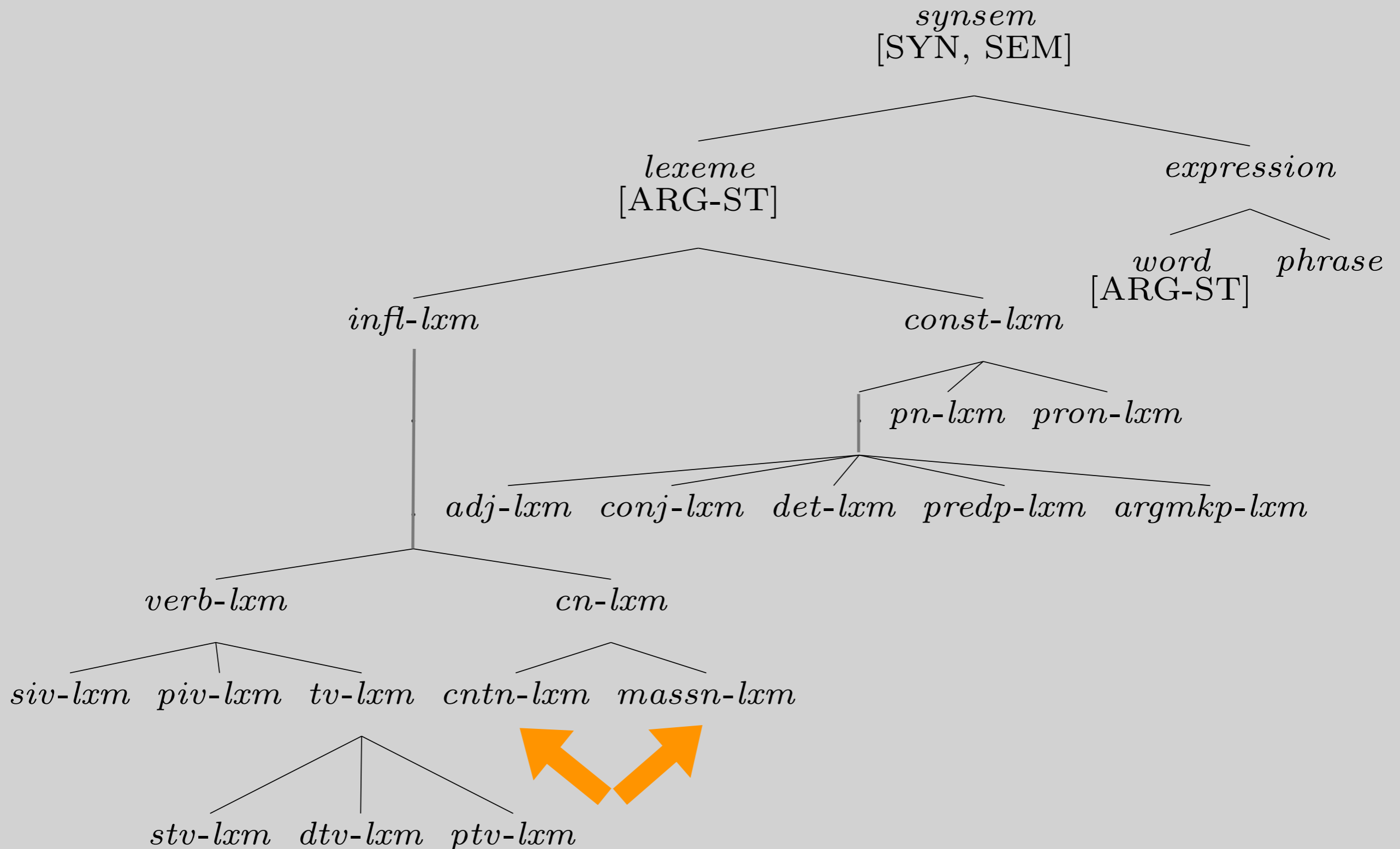


Constraints on *cn-lxm*

cn-lxm :

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

Formally Distinguishing Count vs. Mass Nouns

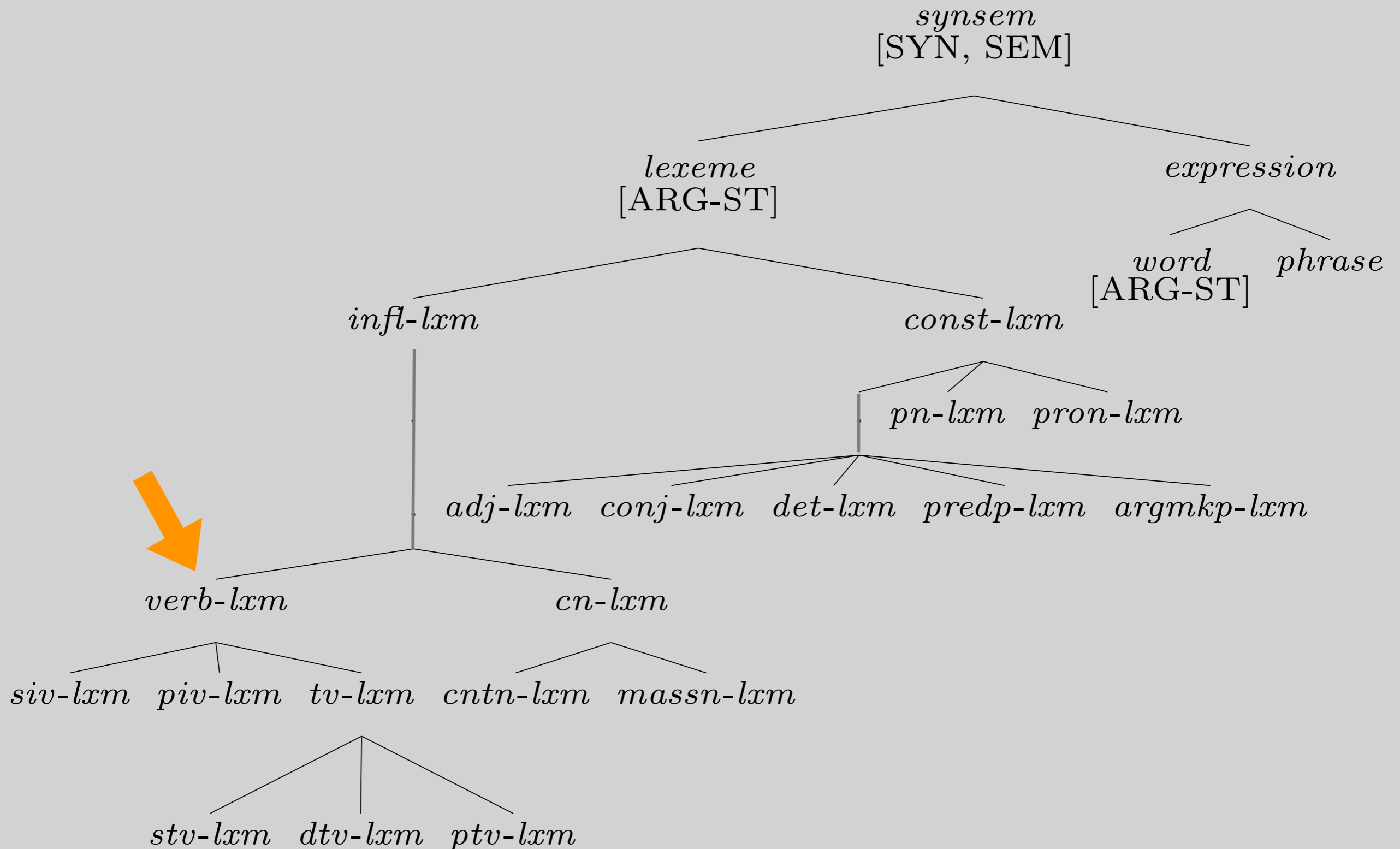


Formally Distinguishing Count vs. Mass Nouns

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

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

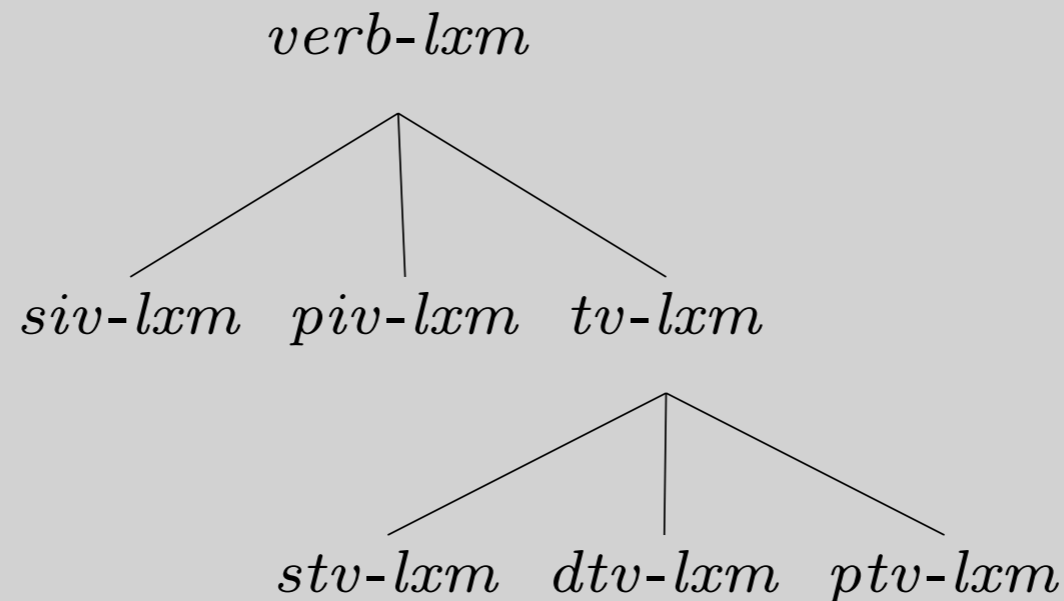
Constraints on *verb-lxm*



Constraints on *verb-lxm*

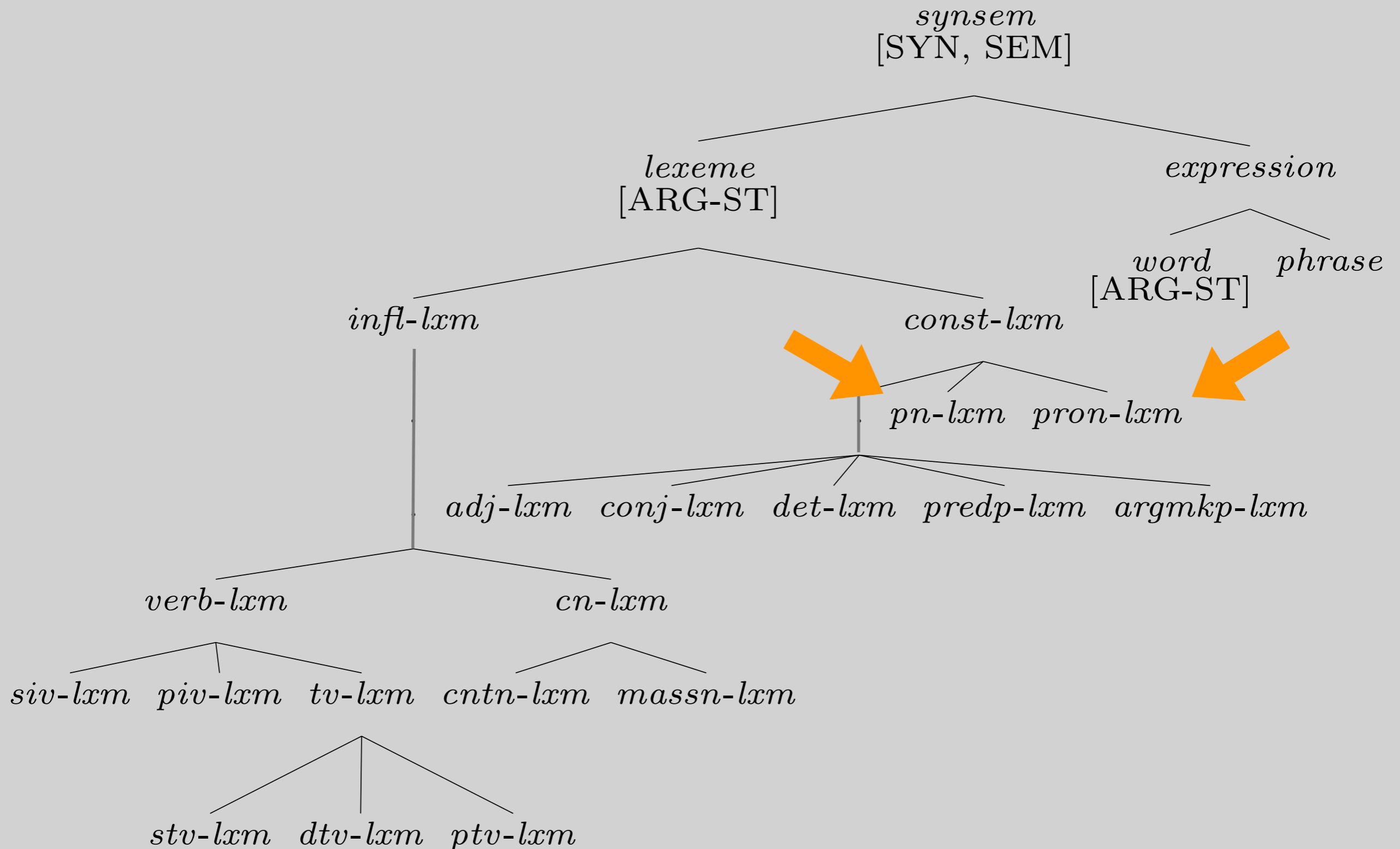
verb-lxm:
$$\left[\begin{array}{l} \text{SYN} \quad \left[\text{HEAD} \quad \textit{verb} \right] \\ \text{SEM} \quad \left[\text{MODE} \quad \textit{prop} \right] \\ \text{ARG-ST} \quad / \langle \text{NP}, \dots \rangle \end{array} \right]$$

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:

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

pron-lxm:

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

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):

$$\begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{IND} & s_0 \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{IND} & s_1 \end{bmatrix} \dots \begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{IND} & s_{n-1} \end{bmatrix} \begin{bmatrix} \text{HEAD} & conj \\ \text{IND} & s_0 \\ \text{RESTR} & \langle \left[\text{ARGS} \langle s_1 \dots s_n \rangle \right] \rangle \end{bmatrix} \begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{IND} & s_n \end{bmatrix}$$

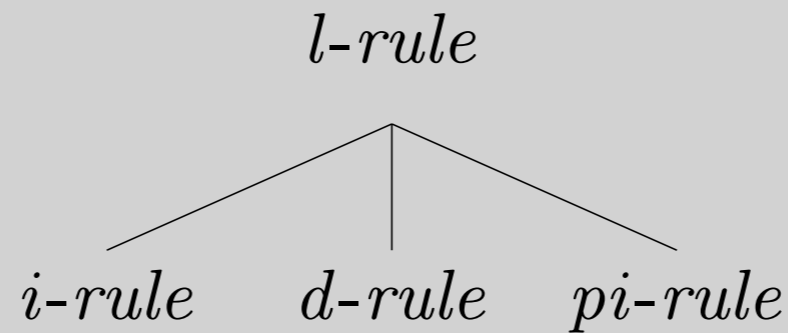
Lexical Types & Lexical Rules

- 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*

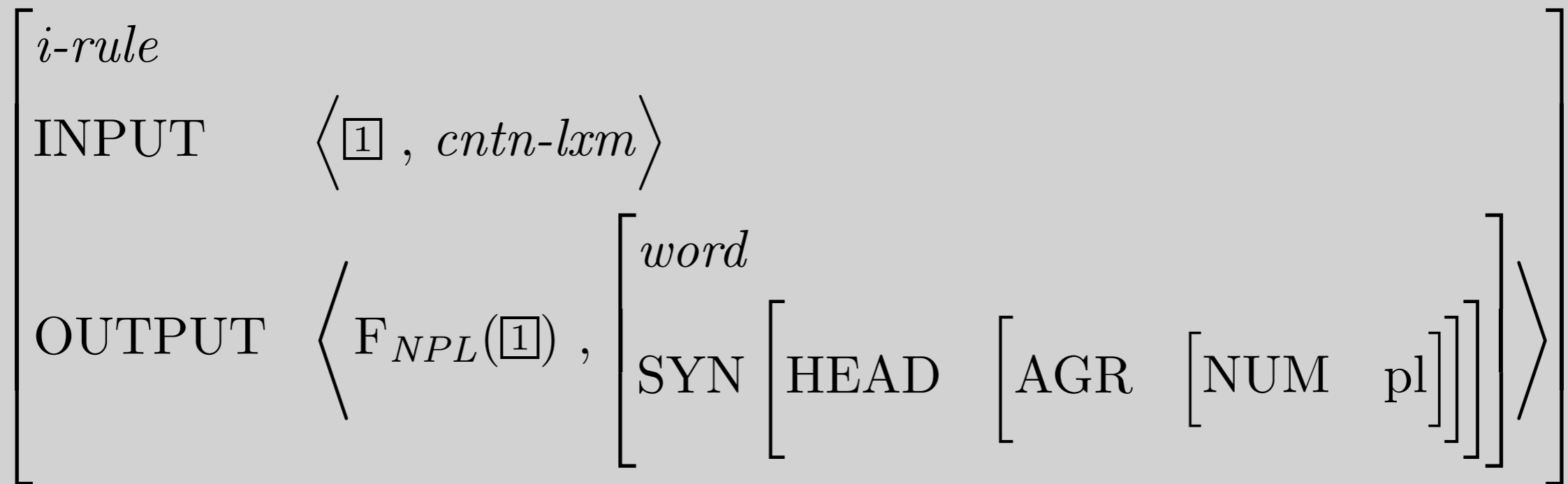


$$l\text{-rule} : \left[\begin{array}{l} \text{INPUT} \quad l\text{-sequence} \left\langle X, [\text{SEM} \quad / \boxed{2}] \right\rangle \\ \text{OUTPUT} \quad l\text{-sequence} \left\langle Y, [\text{SEM} \quad / \boxed{2}] \right\rangle \end{array} \right]$$

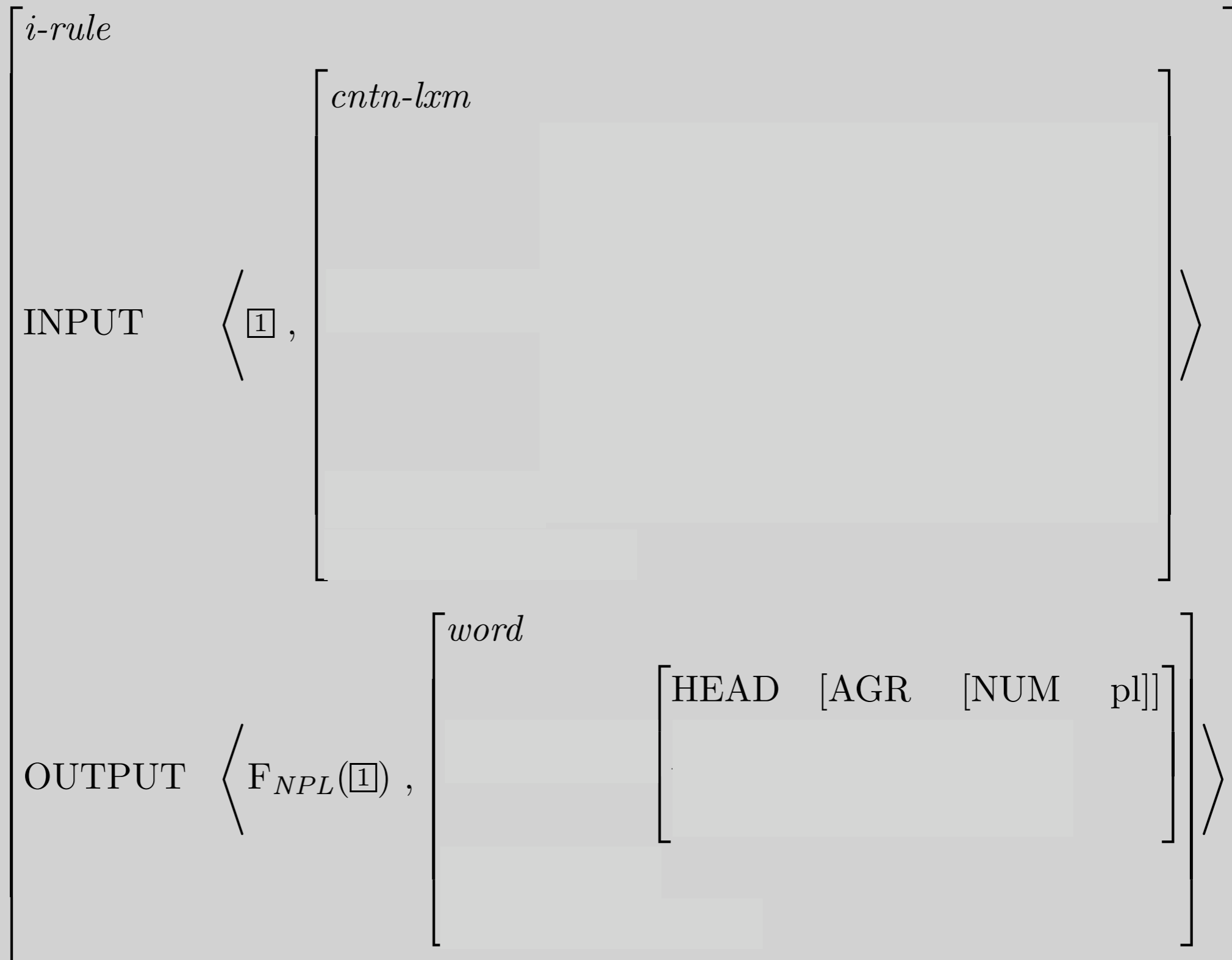
$$i\text{-rule} : \left[\begin{array}{l} \text{INPUT} \quad \left\langle X, \left[\begin{array}{l} \textit{lexeme} \\ \text{SYN} \quad \boxed{3} \\ \text{ARG-ST} \quad \boxed{A} \end{array} \right] \right\rangle \\ \text{OUTPUT} \quad \left\langle Y, \left[\begin{array}{l} \textit{word} \\ \text{SYN} \quad \boxed{3} \\ \text{ARG-ST} \quad \boxed{A} \end{array} \right] \right\rangle \end{array} \right]$$

$$d\text{-rule} : \left[\begin{array}{l} \text{INPUT} \quad \left\langle X, \left[\begin{array}{l} \textit{lexeme} \\ \text{SYN} \quad / \boxed{3} \end{array} \right] \right\rangle \\ \text{OUTPUT} \quad \left\langle Y, \left[\begin{array}{l} \textit{lexeme} \\ \text{SYN} \quad / \boxed{3} \end{array} \right] \right\rangle \end{array} \right]$$

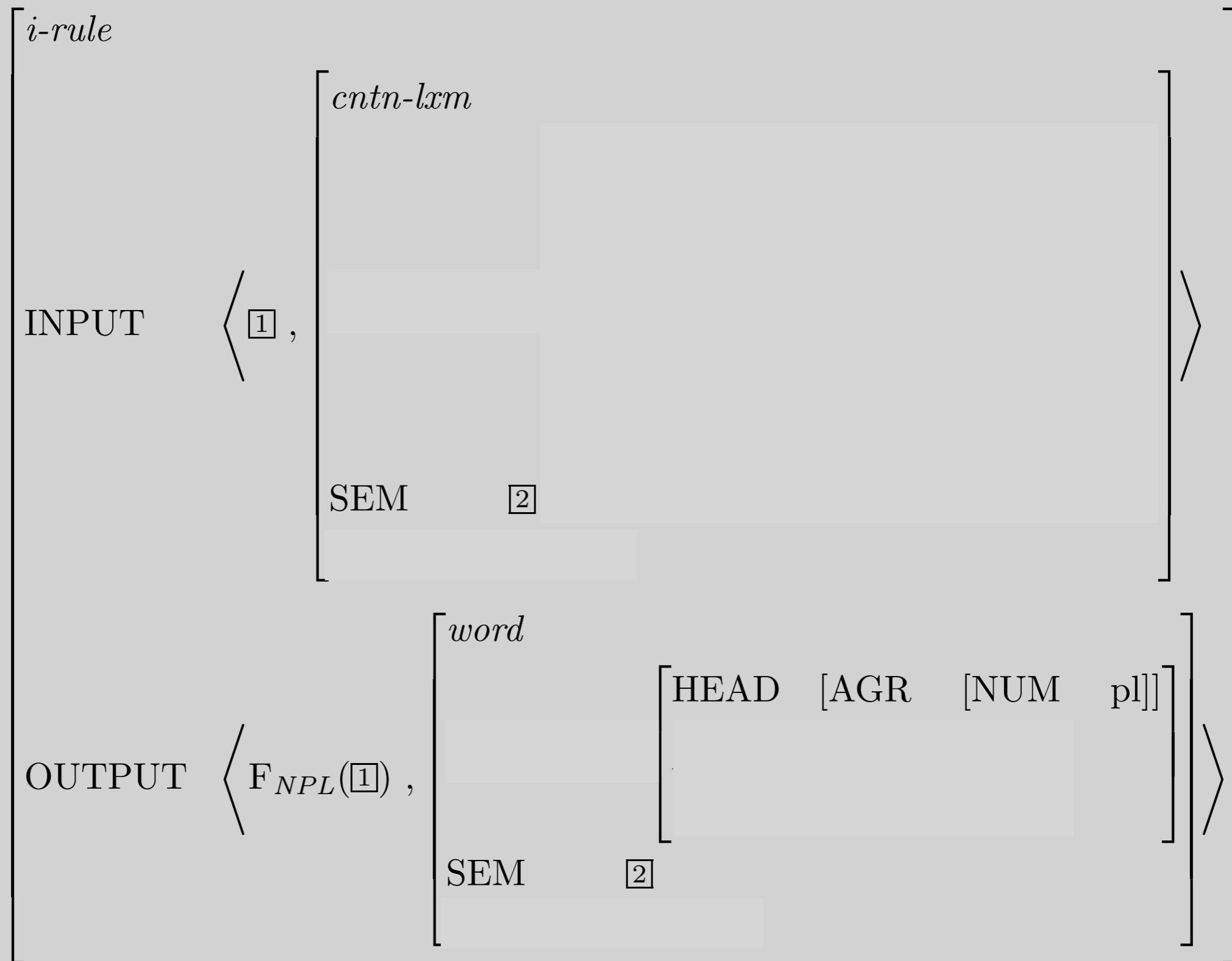
Plural Noun LR



Plural Noun LR



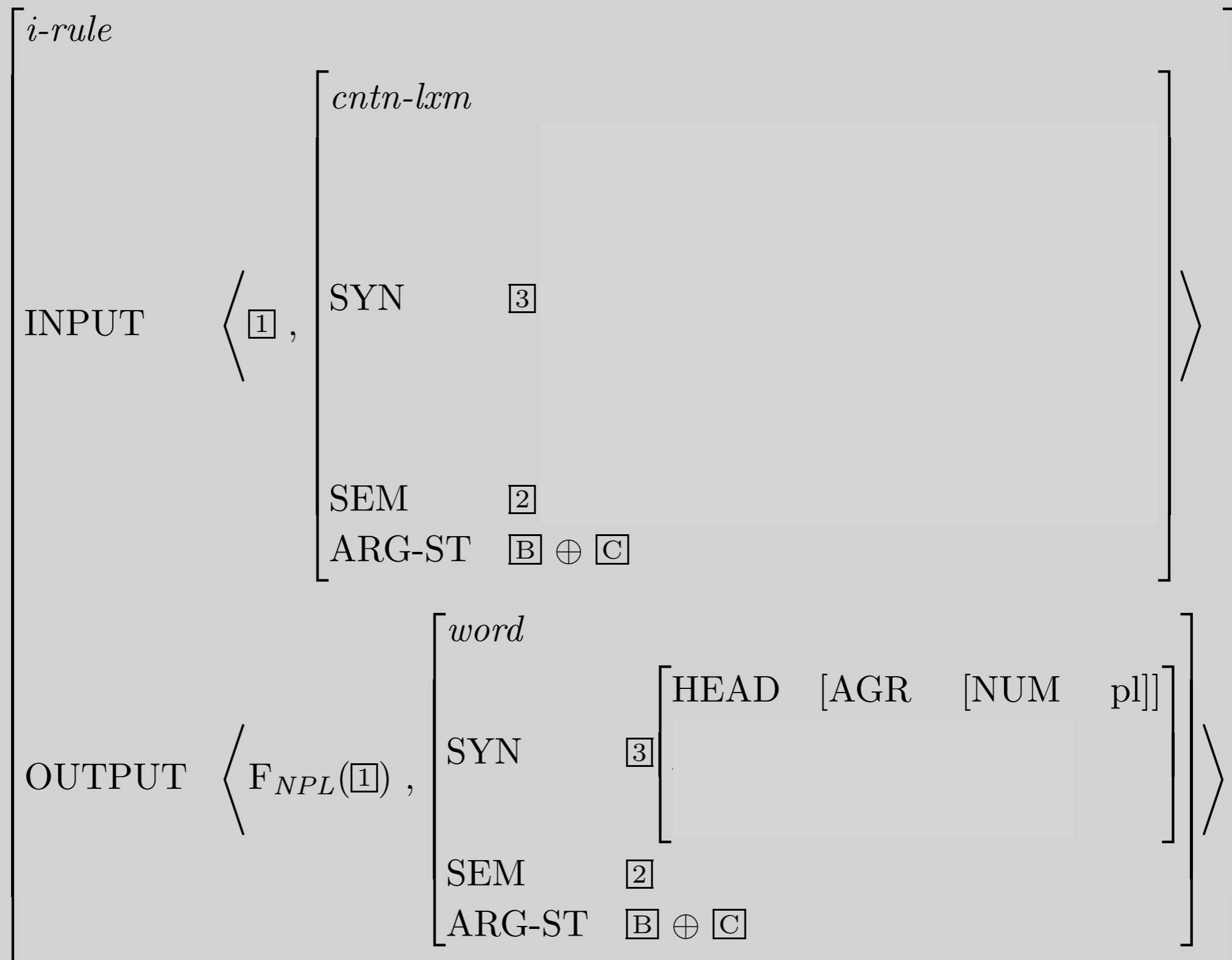
Plural Noun LR with Inherited Constraints



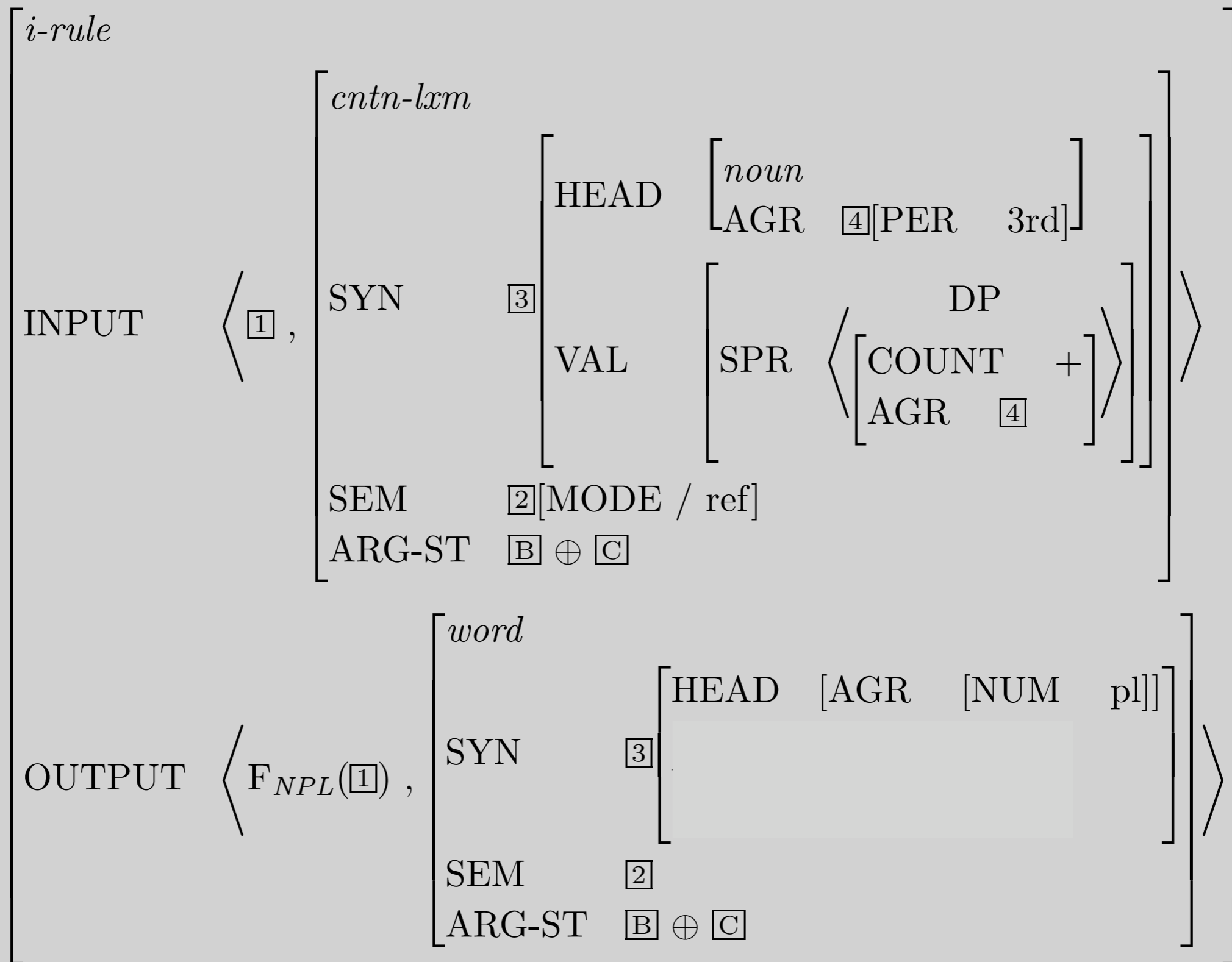
Plural Noun LR with Inherited Constraints



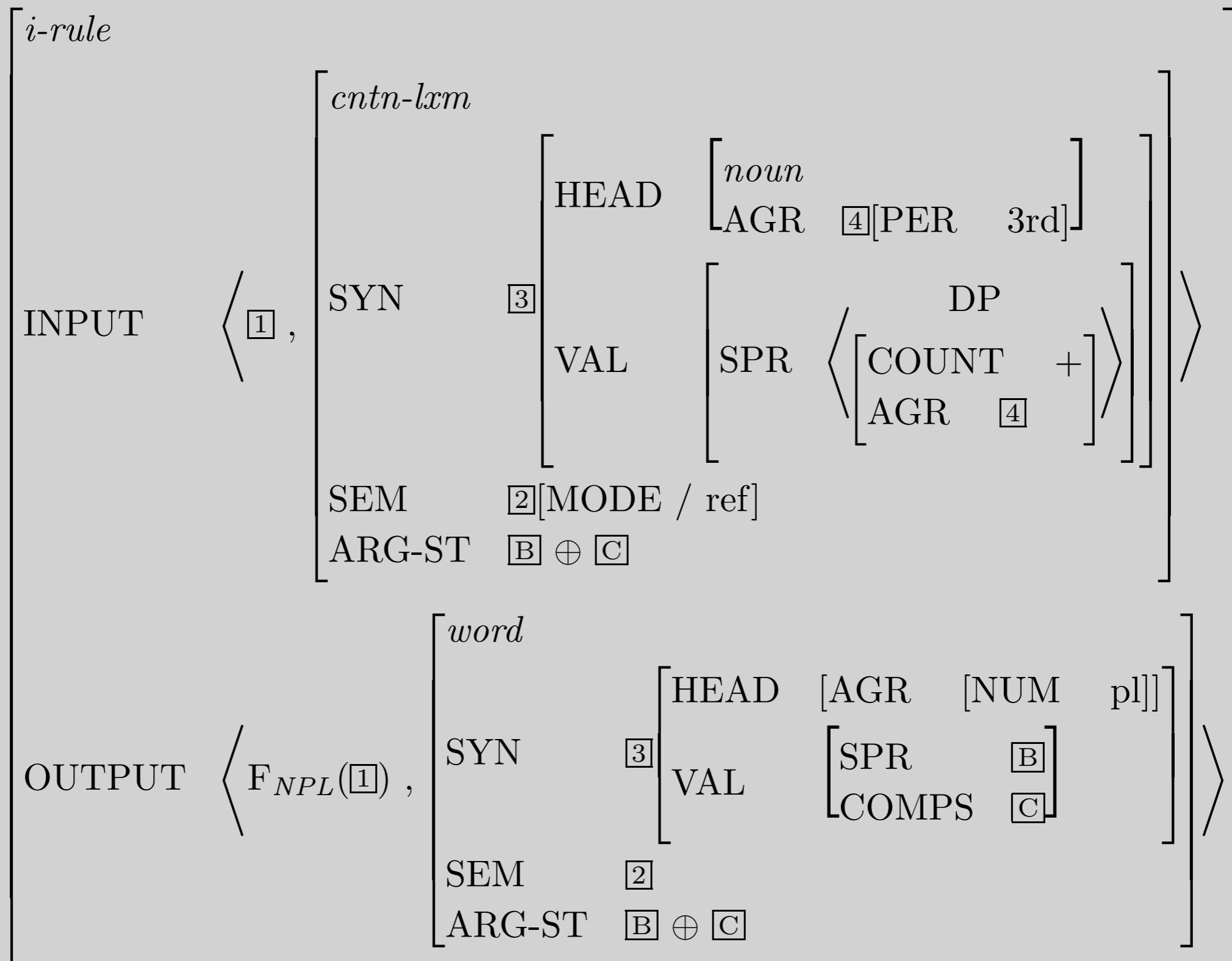
Plural Noun LR with Inherited Constraints



Plural Noun LR with Inherited Constraints



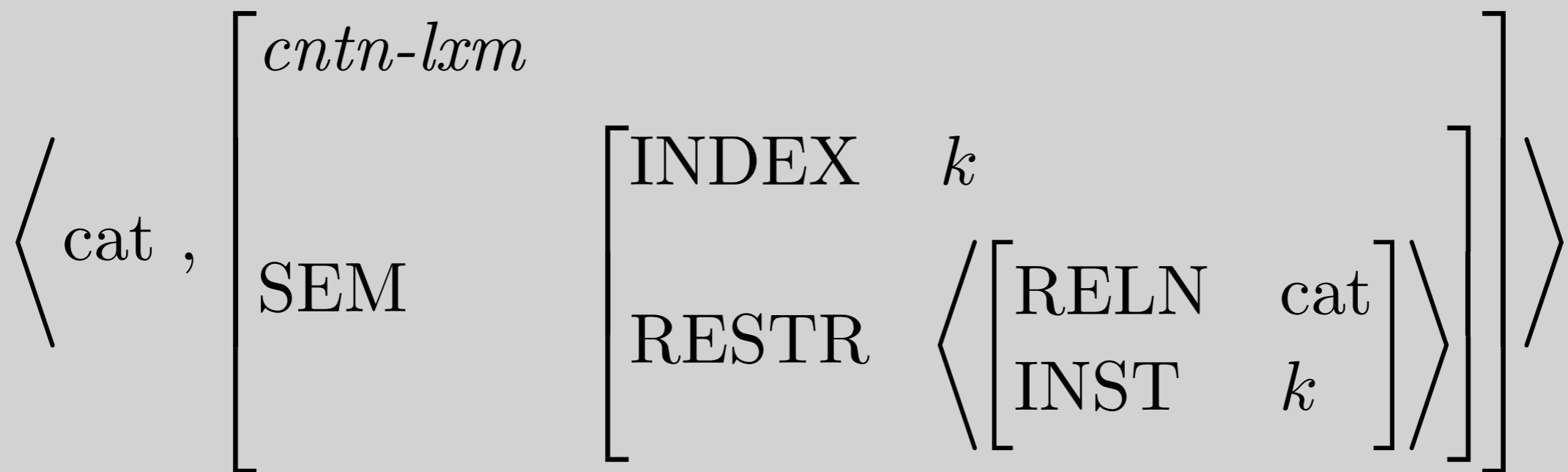
Plural Noun LR with Inherited Constraints



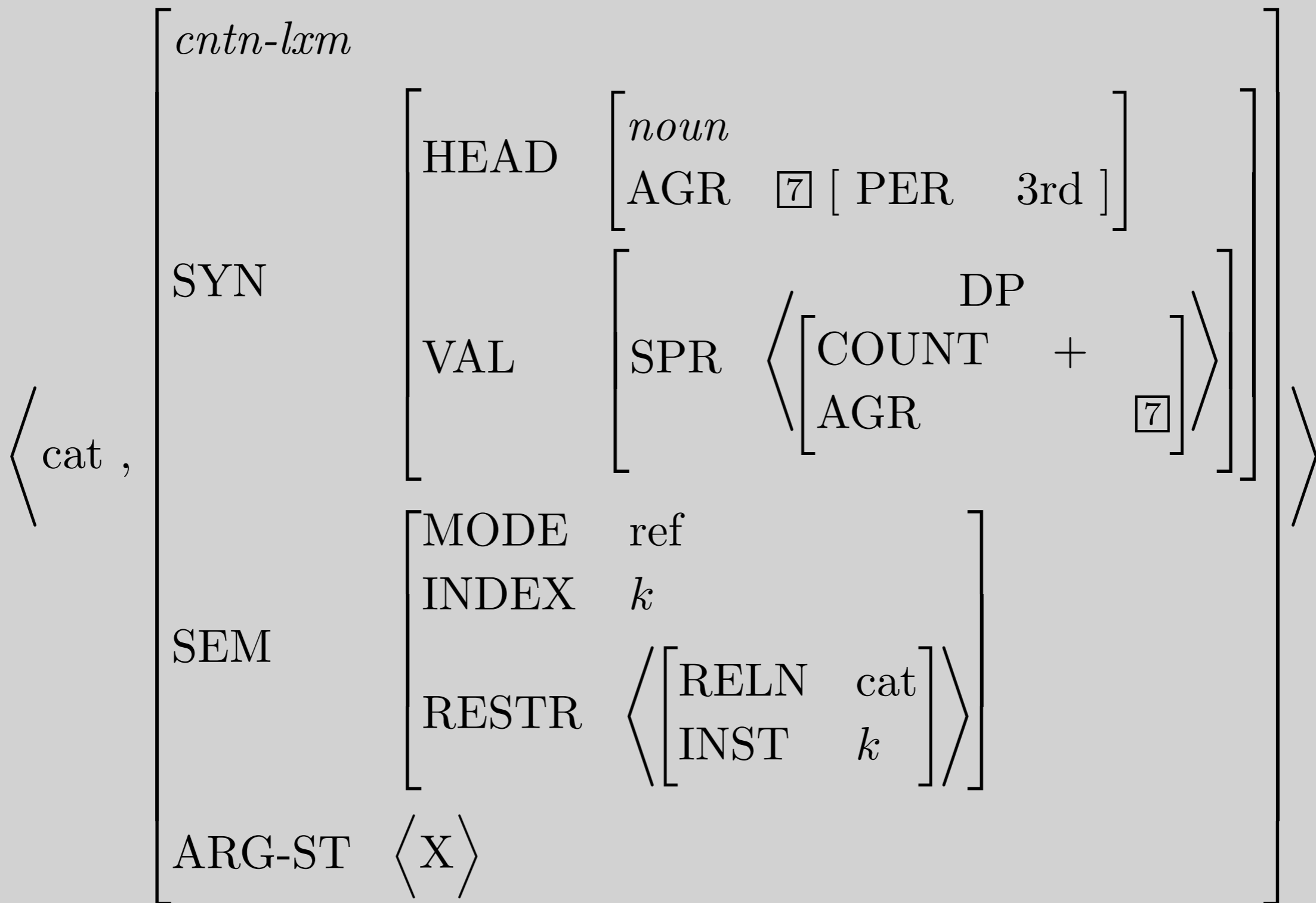
Practicalities - Applying Lexical Rules

- 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 unconstrained.

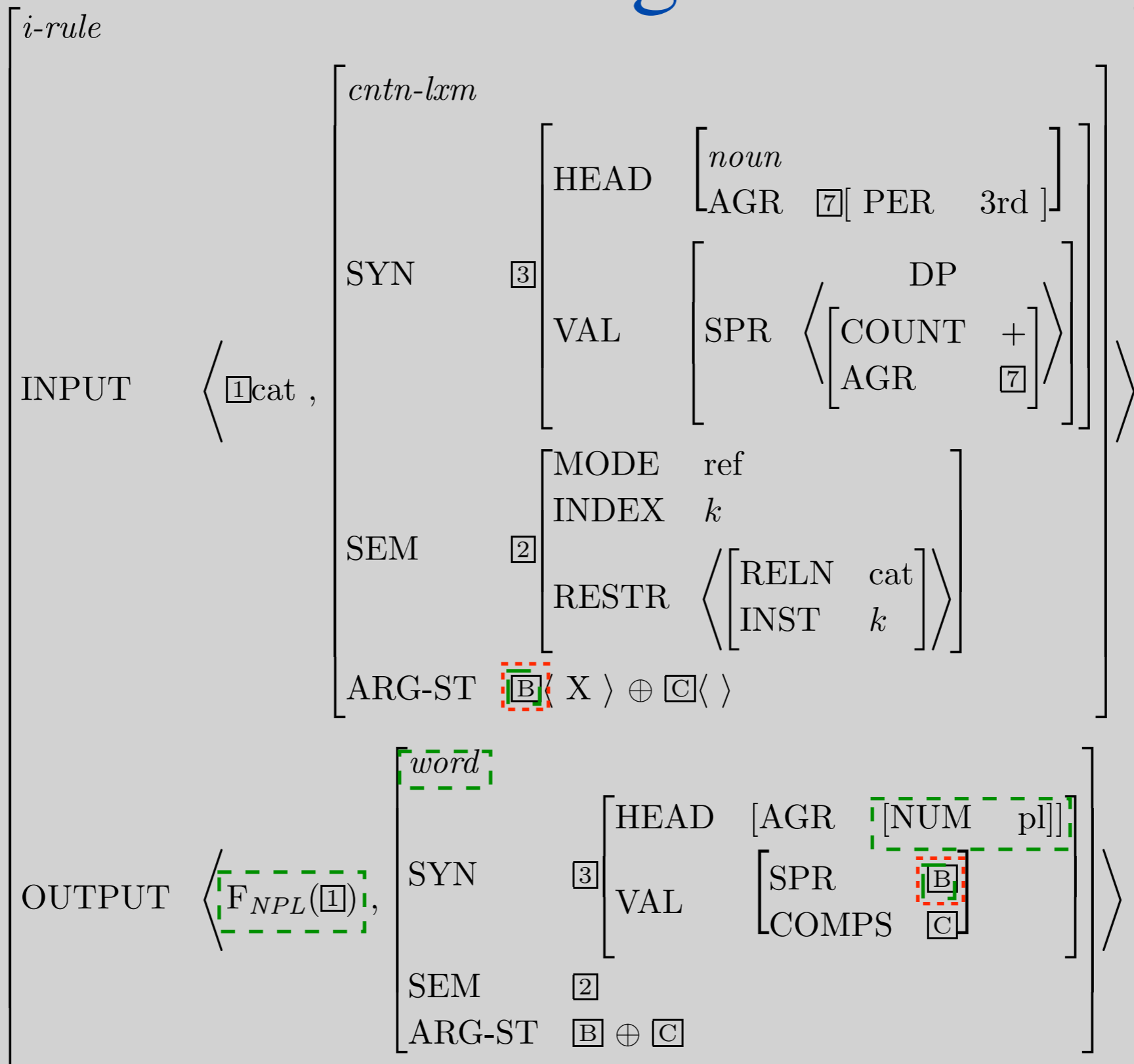
Example: Lexical Entry for *cat*



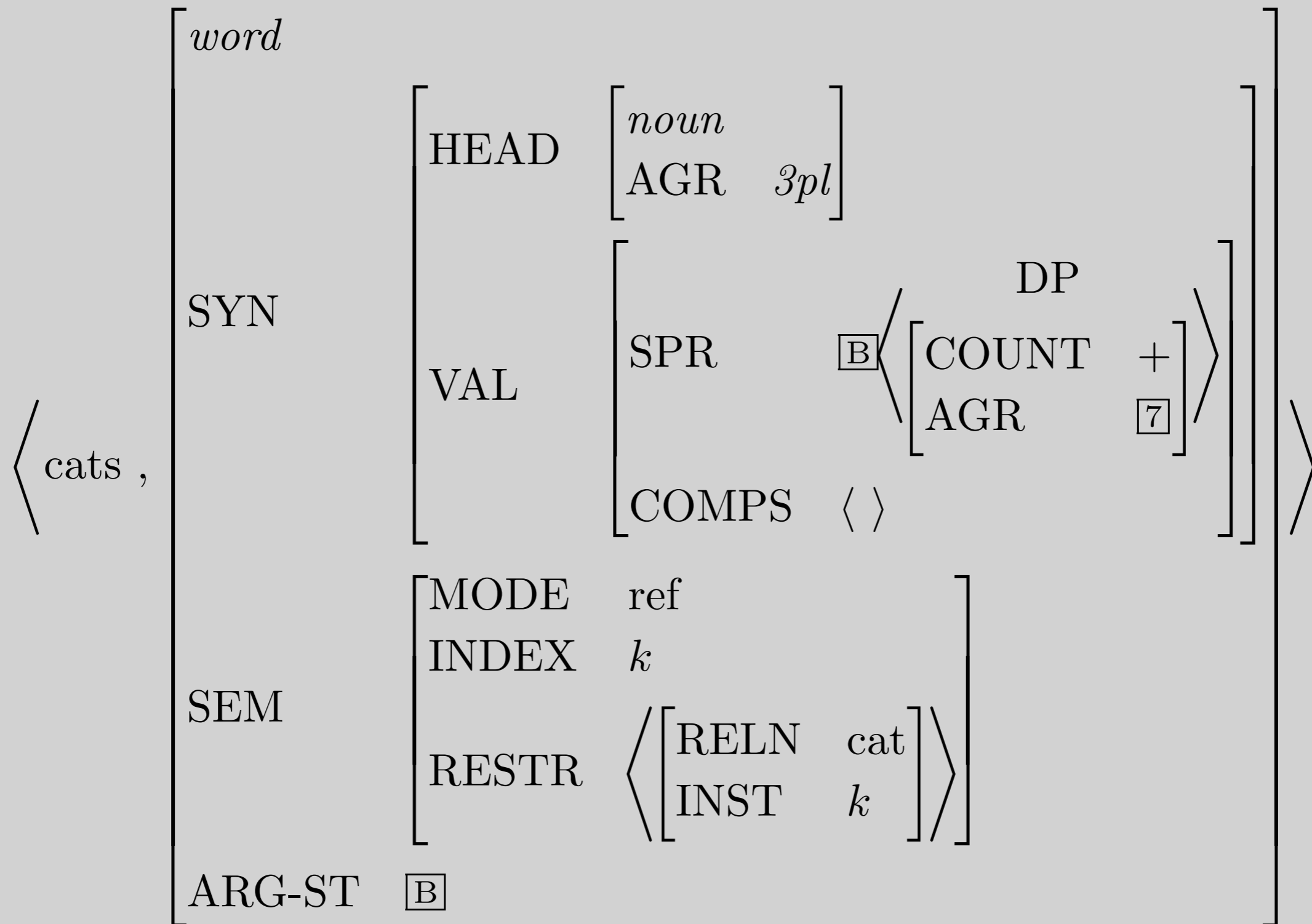
Example: *cat*, with inheritance



Licensing *cats*



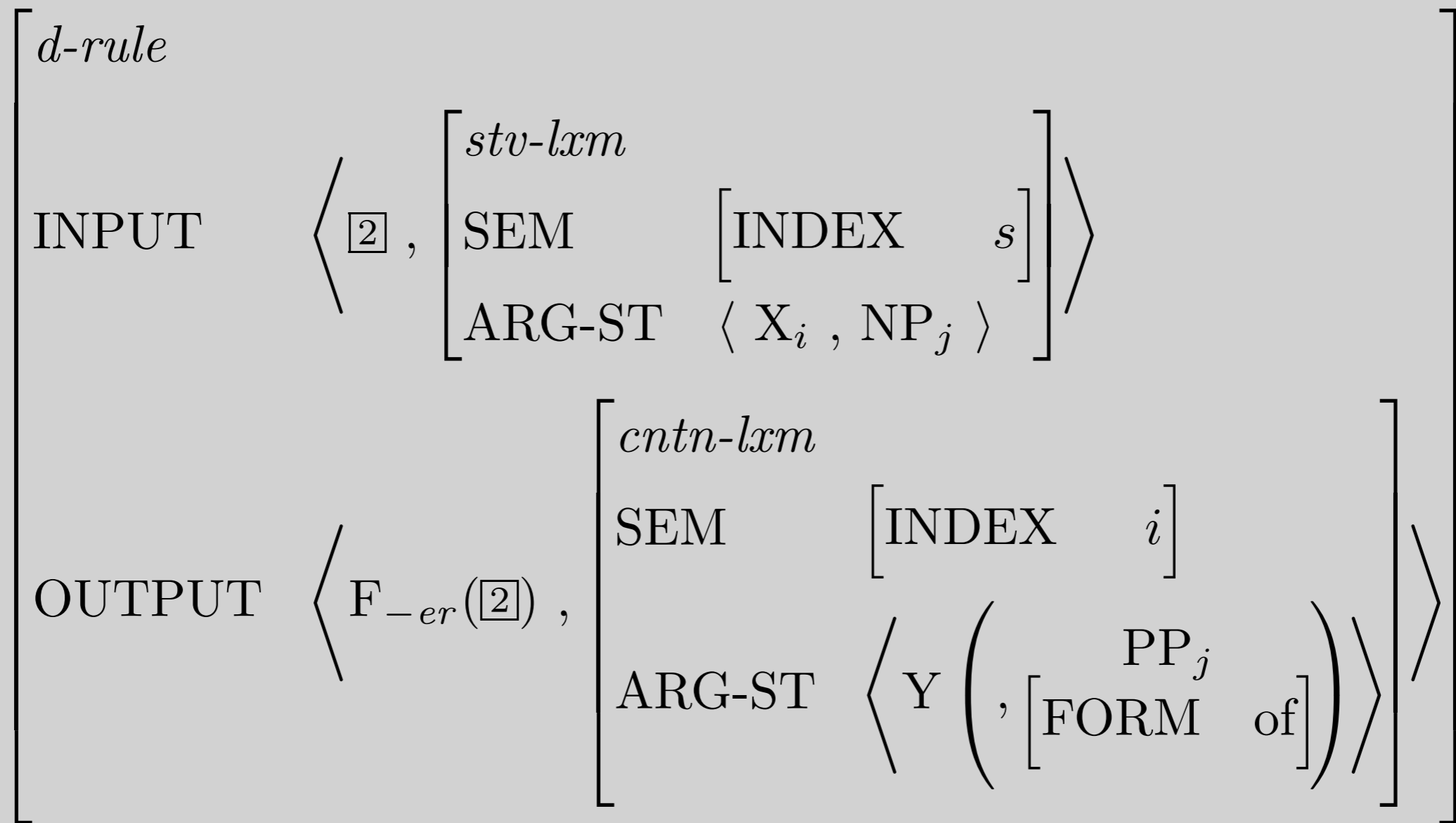
cats: The Lexical Sequence



Derivational Lexical Rules

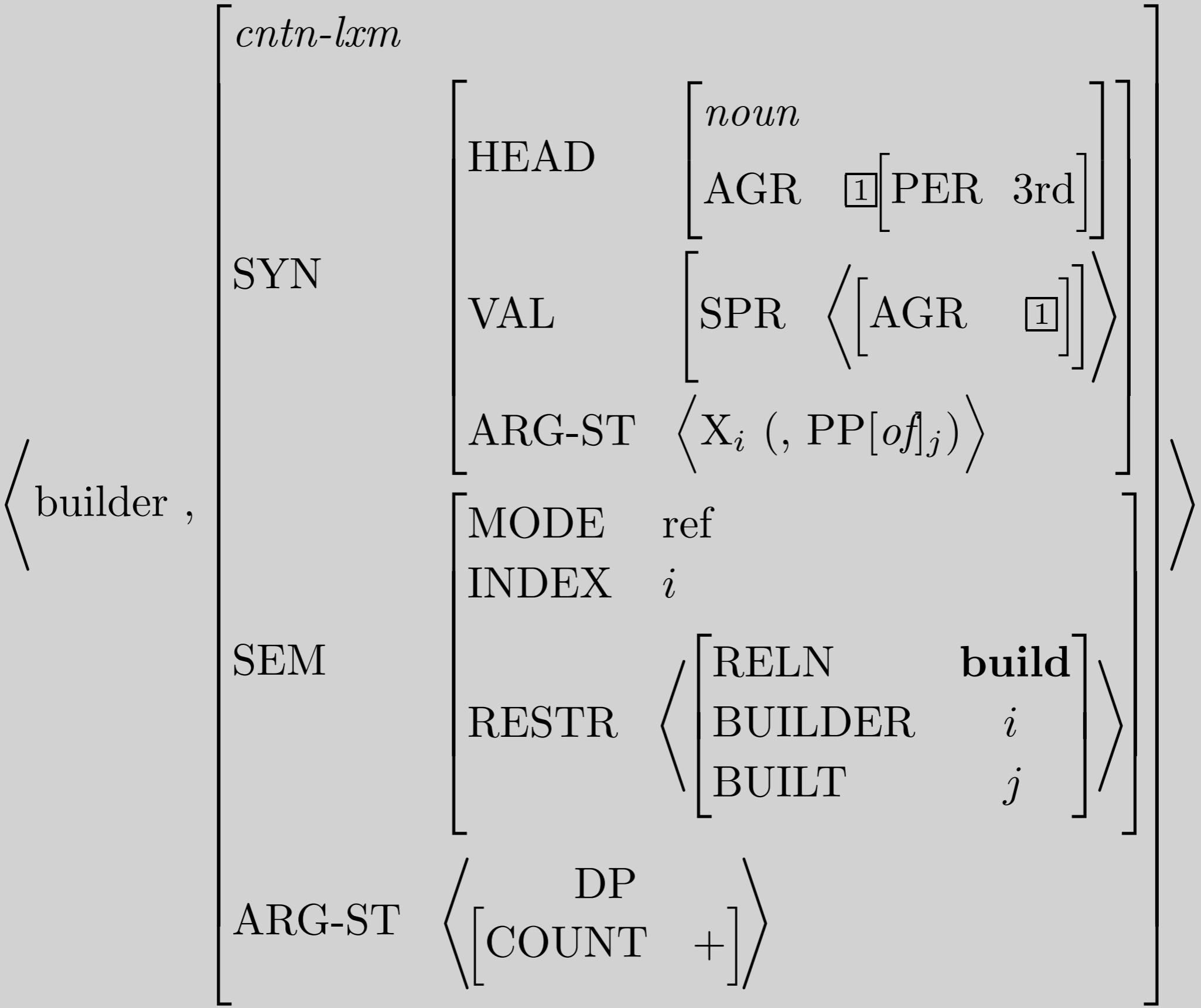
$$d\text{-rule} : \left[\begin{array}{l} \text{INPUT} \\ \text{OUTPUT} \end{array} \left\langle X, \left[\begin{array}{l} \textit{lexeme} \\ \text{SYN} \end{array} \right] / \boxed{3} \right\rangle \right]$$

A Derivational LR: Agent Nominalization



the discoverer of oxygen
a builder of bridges

**a relier on Sandy*
**the putter of beer in the fridge*



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

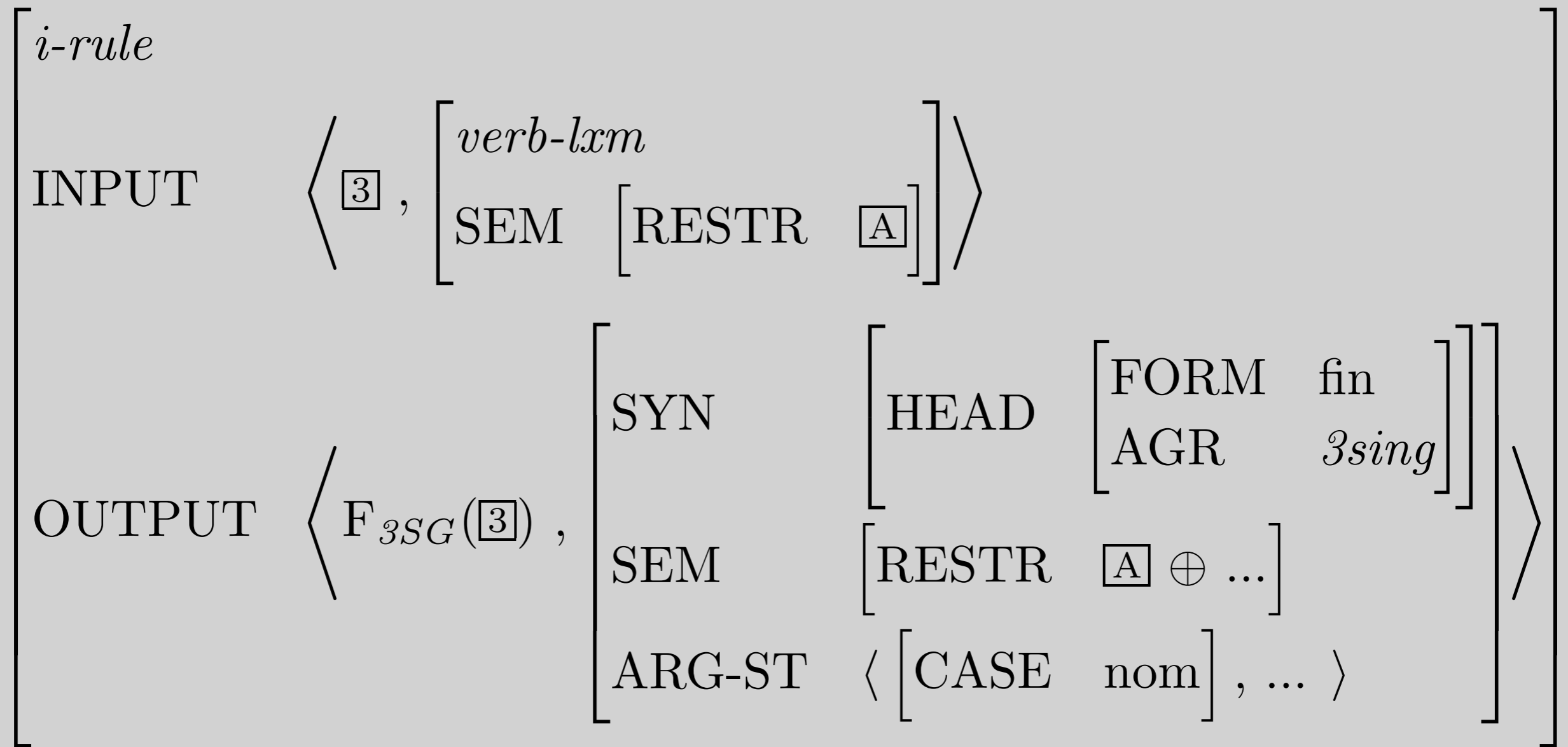
Singular Noun Lexical Rule

$$\left[\begin{array}{l}
 \textit{i-rule} \\
 \text{INPUT} \quad \langle \boxed{1}, \textit{cn-lxm} \rangle \\
 \text{OUTPUT} \quad \langle \boxed{1}, \left[\text{SYN} \left[\text{HEAD} \left[\text{AGR} \left[\text{NUM sg} \right] \right] \right] \right] \rangle
 \end{array} \right]$$

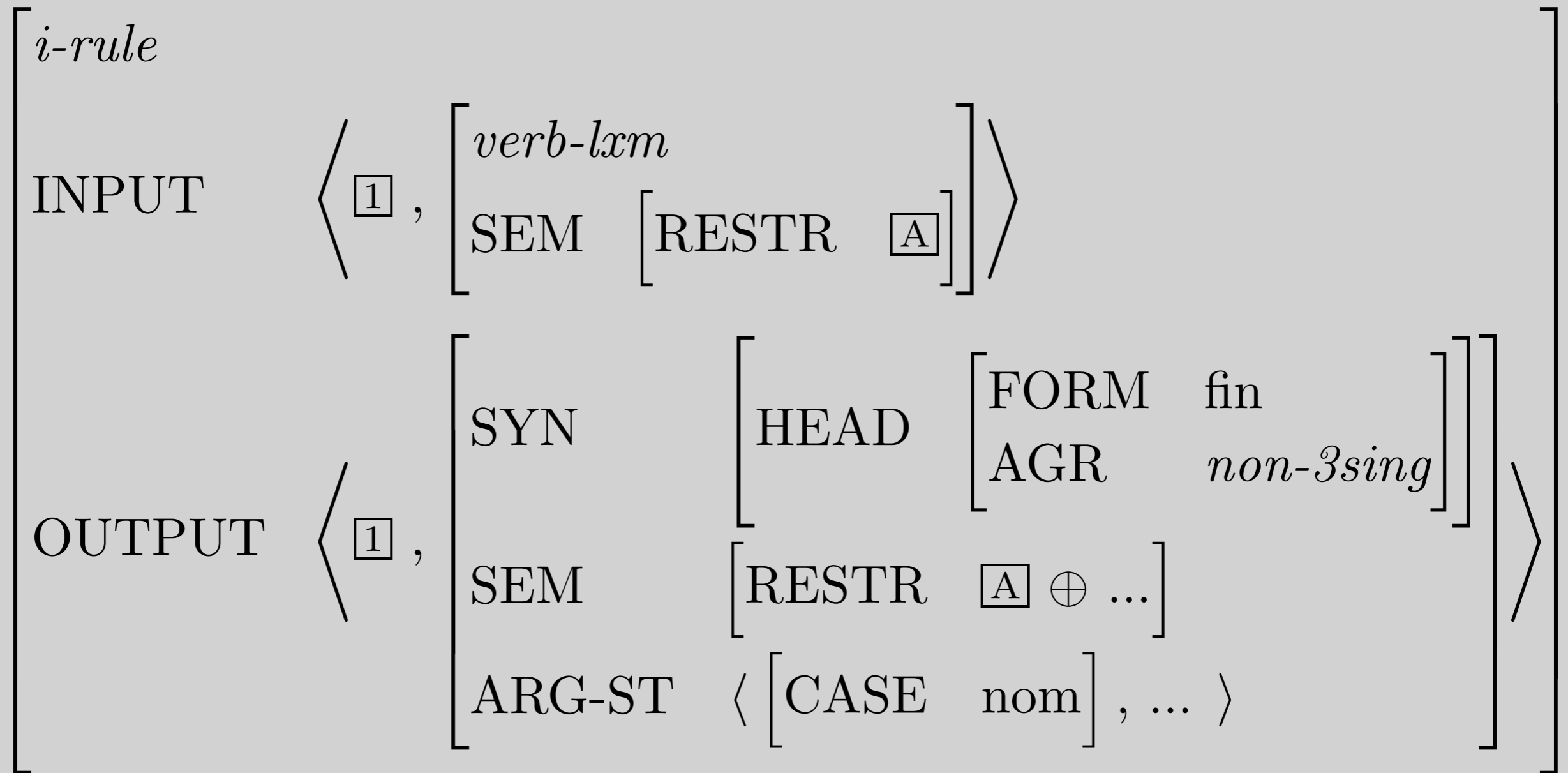
Plural Noun Lexical Rule

$$\left[\begin{array}{l}
 \textit{i-rule} \\
 \text{INPUT} \quad \langle \boxed{1}, \textit{cntn-lxm} \rangle \\
 \text{OUTPUT} \quad \langle F_{NPL}(\boxed{1}), \left[\text{SYN} \left[\text{HEAD} \left[\text{AGR} \left[\text{NUM pl} \right] \right] \right] \right] \rangle
 \end{array} \right]$$

3rd-Singular Verb Lexical Rule



Non-3rd-Singular Verb Lexical Rule



Past Tense Lexical Rule

$$\left[\begin{array}{l}
 \textit{i-rule} \\
 \text{INPUT} \quad \left\langle \boxed{3}, \left[\textit{verb-lxm} \right. \right. \\
 \quad \quad \quad \left. \left. \text{SEM} \left[\text{RESTR} \quad \boxed{A} \right] \right] \right\rangle \\
 \\
 \text{OUTPUT} \quad \left\langle F_{\textit{PAST}}(\boxed{3}), \left[\begin{array}{l}
 \text{SYN} \quad \left[\text{HEAD} \quad \left[\text{FORM} \quad \textit{fin} \right] \right] \\
 \text{SEM} \quad \left[\text{RESTR} \quad \boxed{A} \oplus \dots \right] \\
 \text{ARG-ST} \quad \left\langle \left[\text{CASE} \quad \textit{nom} \right], \dots \right\rangle
 \end{array} \right] \right\rangle
 \end{array} \right]$$

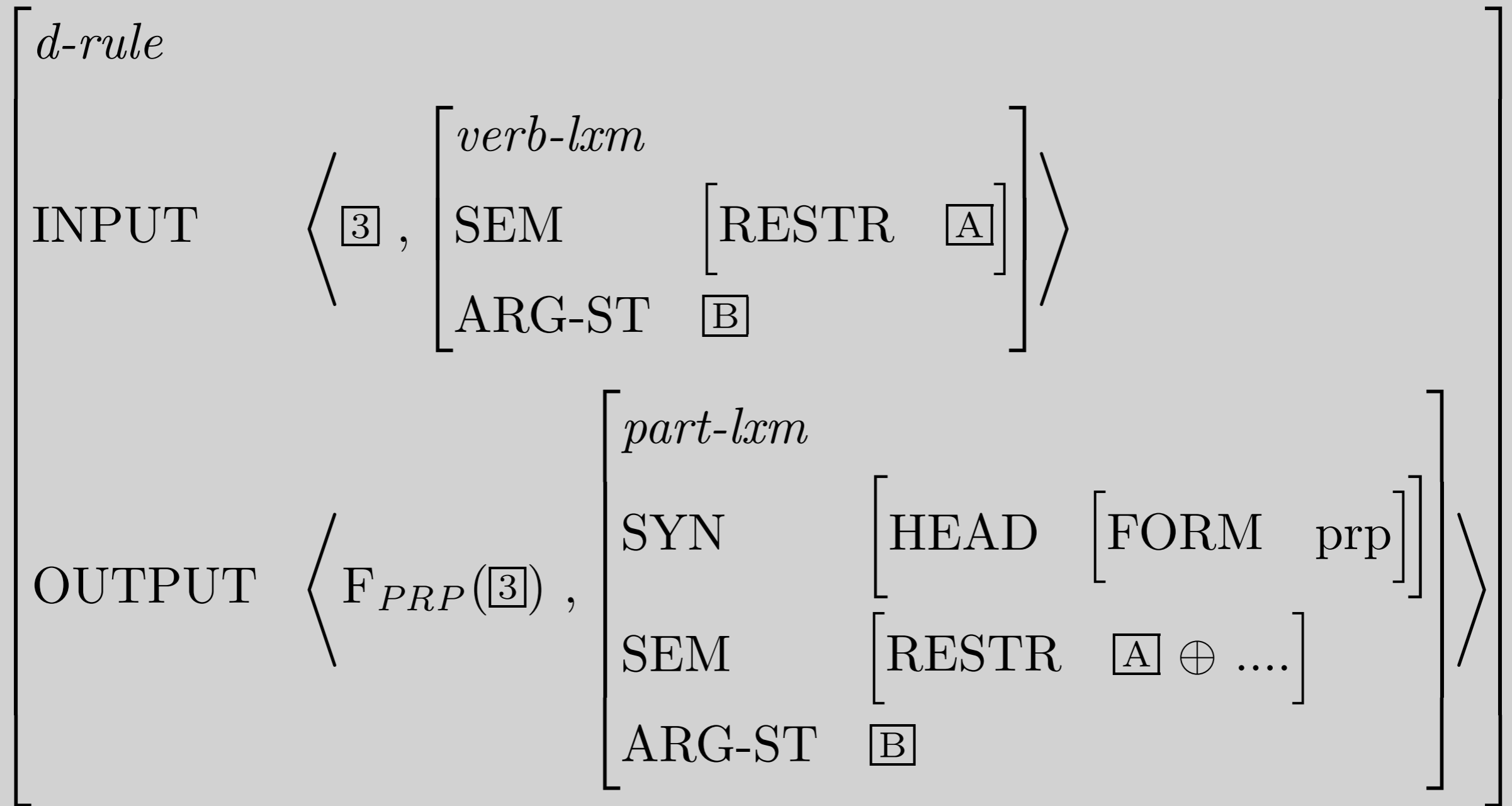
Base Form Lexical Rule

$$\left[\begin{array}{l} i\text{-rule} \\ \text{INPUT} \quad \langle \boxed{1}, \textit{verb-lxm} \rangle \\ \text{OUTPUT} \quad \langle \boxed{1}, \left[\text{SYN} \left[\text{HEAD} \left[\text{FORM} \quad \text{base} \right] \right] \right] \rangle \end{array} \right]$$

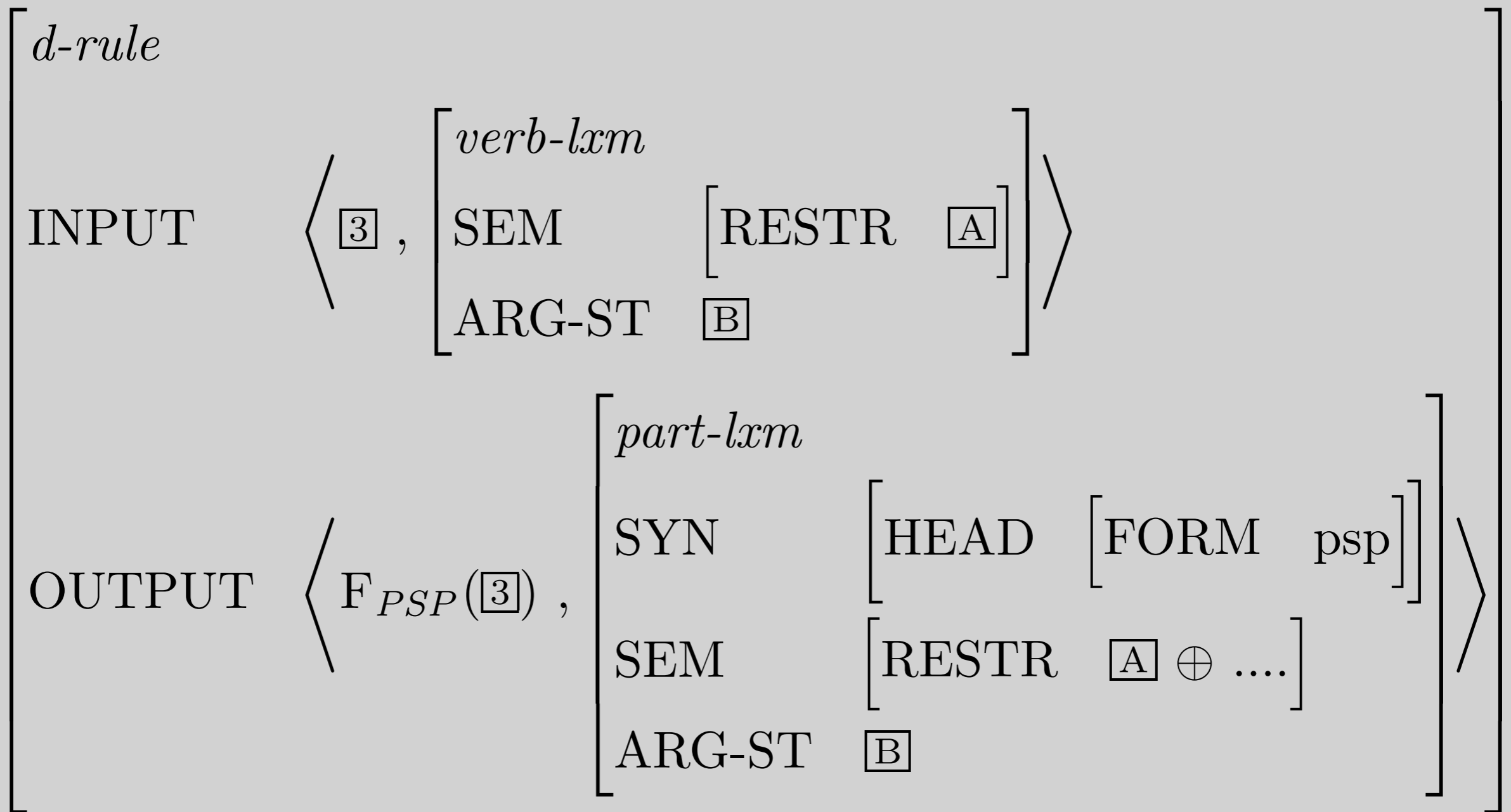
Constant Lexeme Lexical Rule

$$\left[\begin{array}{l} i\text{-rule} \\ \text{INPUT} \quad \langle \boxed{1}, \textit{const-lxm} \rangle \\ \text{OUTPUT} \quad \left[\text{FIRST} \quad \boxed{1} \right] \end{array} \right]$$

Present Participle Lexical Rule



Past Participle Lexical Rule



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).