

Ling 566

Oct 15, 2013

How the Grammar Works

Overview

- What we're trying to do
- The pieces of our grammar
- Two extended examples
- Reflection on what we've done, what we still have to do
- Reading questions

What We're Trying To Do

- Objectives
 - Develop a theory of knowledge of language
 - Represent linguistic information explicitly enough to distinguish well-formed from ill-formed expressions
 - Be parsimonious, capturing linguistically significant generalizations.
- Why Formalize?
 - To formulate testable predictions
 - To check for consistency
 - To make it possible to get a computer to do it for us

How We Construct Sentences

- The Components of Our Grammar
 - Grammar rules
 - Lexical entries
 - Principles
 - Type hierarchy (very preliminary, so far)
 - Initial symbol (S, for now)
- We combine constraints from these components.
 - Q: What says we have to combine them?

An Example

A cat slept.

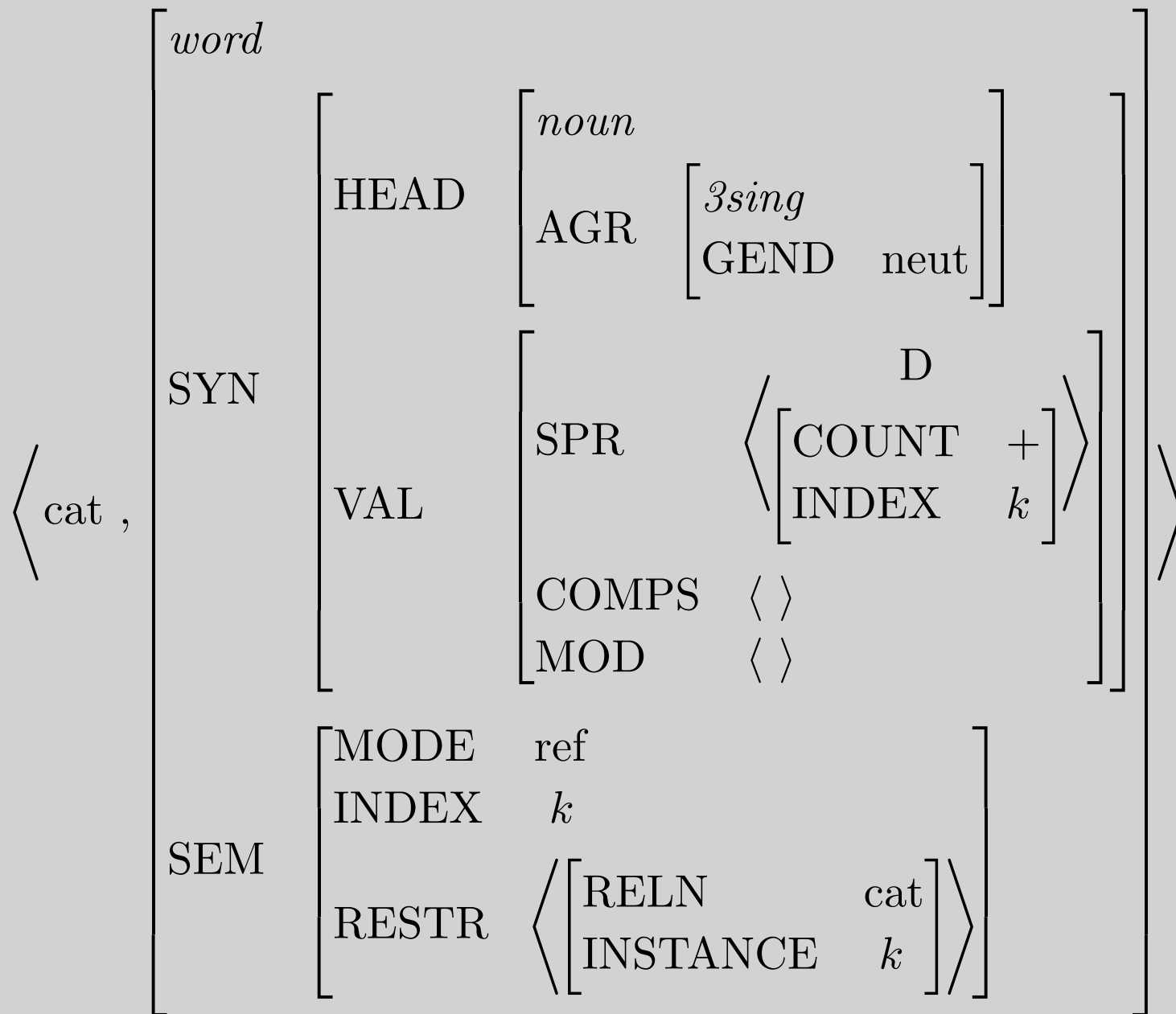
- Can we build this with our tools?
- Given the constraints our grammar puts on well-formed sentences, is this one?

Lexical Entry for *a*

$\langle a, \rangle$	$\left[\begin{array}{l} \text{word} \\ \\ \text{SYN} \\ \\ \text{SEM} \end{array} \right.$	$\left[\begin{array}{l} \text{HEAD} \\ \\ \text{VAL} \\ \\ \text{MODE} \\ \text{INDEX} \\ \text{RESTR} \end{array} \right.$	$\left[\begin{array}{l} \text{det} \\ \text{AGR} \\ \text{COUNT} \\ \\ \text{COMPS} \\ \text{SPR} \\ \text{MOD} \\ \\ \text{none} \\ j \\ \left\langle \left[\begin{array}{l} \text{RELN} \\ \text{BV} \end{array} \right] \right\rangle \end{array} \right.$	$\left[\begin{array}{l} \\ \\ 3sing \\ + \\ \langle \rangle \\ \langle \rangle \\ \langle \rangle \\ \\ \\ a \\ j \end{array} \right.$	\rangle

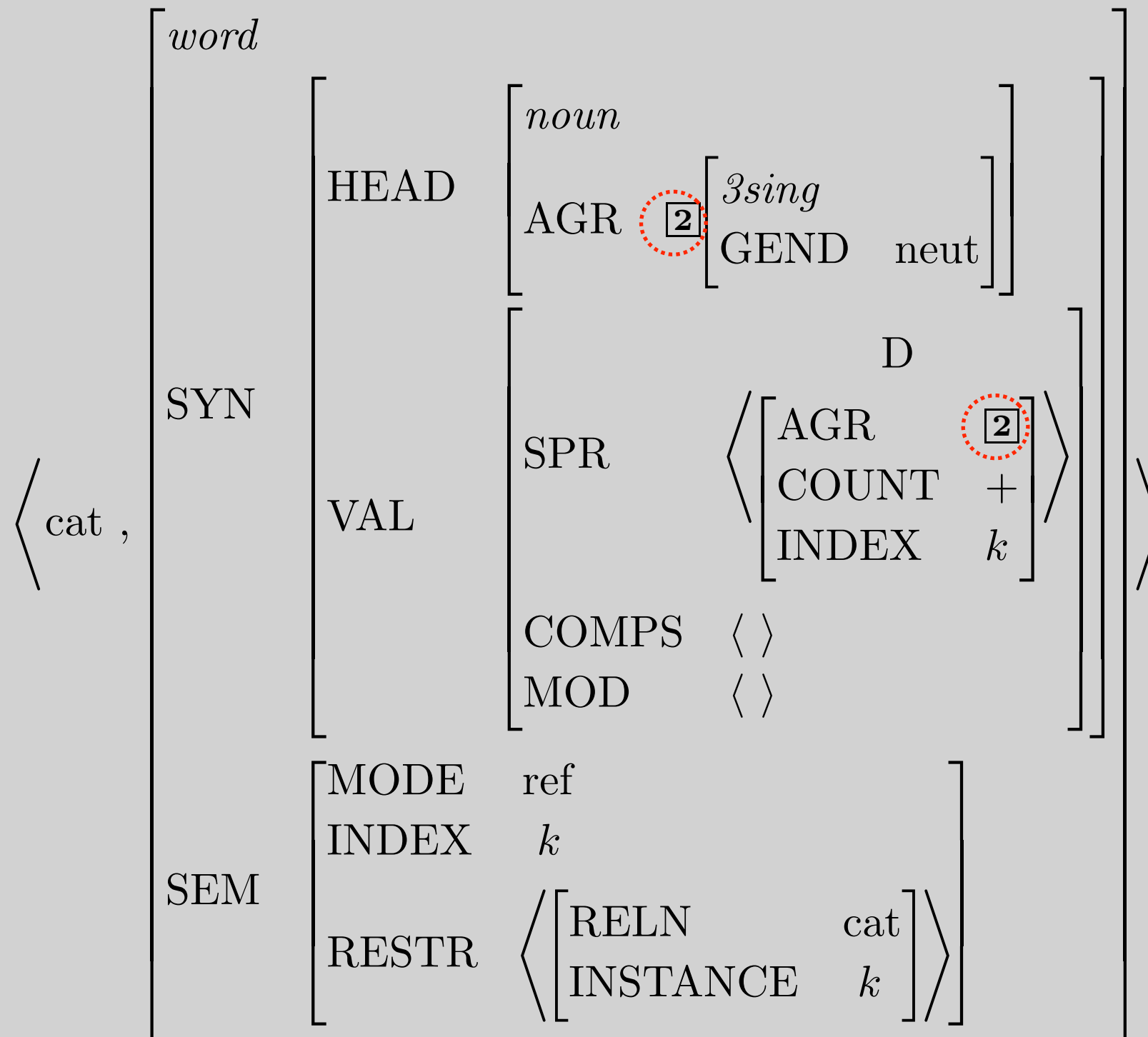
- Is this a fully specified description?
- What features are unspecified?
- How many word structures can this entry license?

Lexical Entry for *cat*

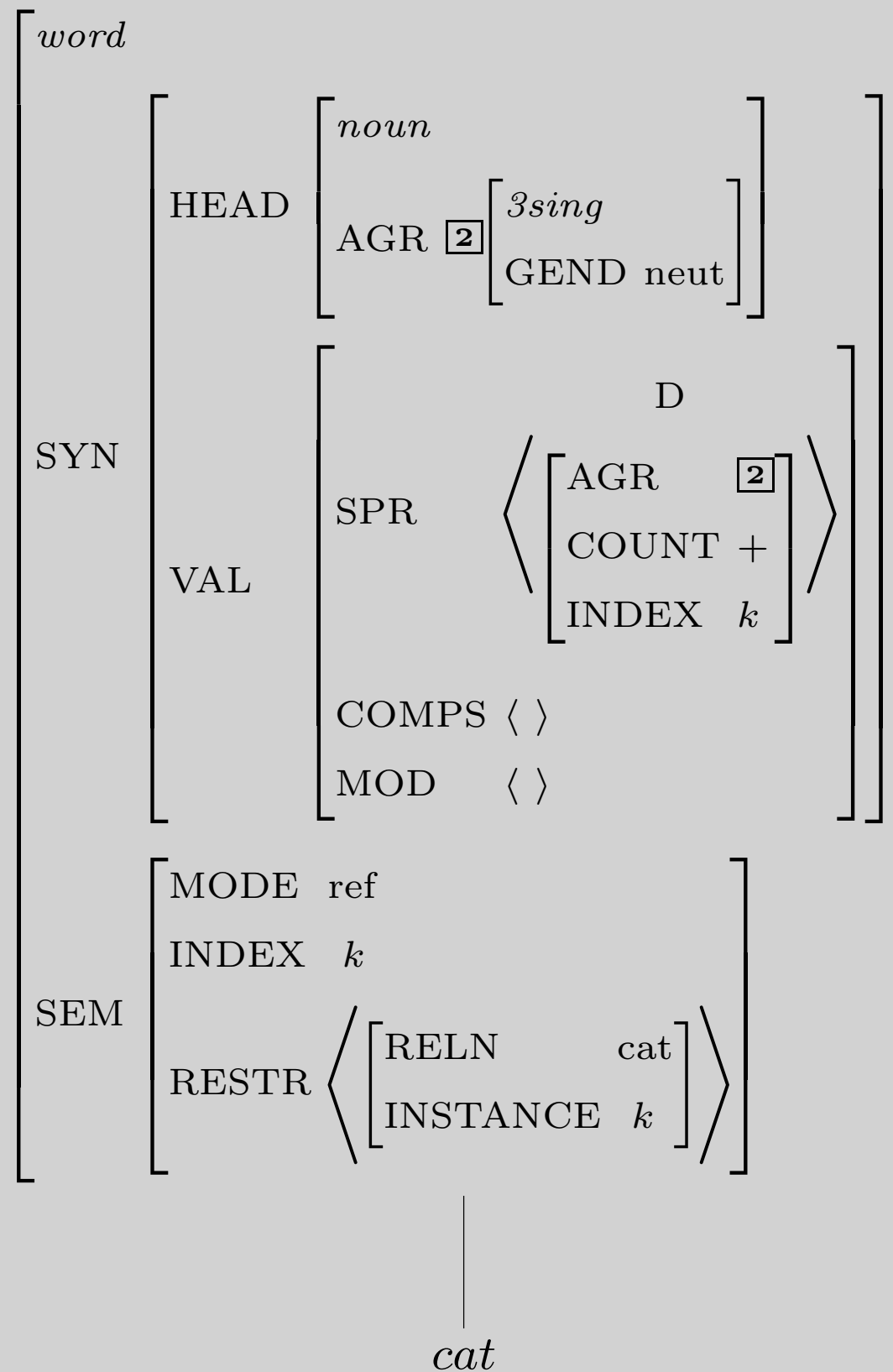


- Which feature paths are abbreviated?
- Is this a fully specified description?
- What features are unspecified?
- How many word structures can this entry license?

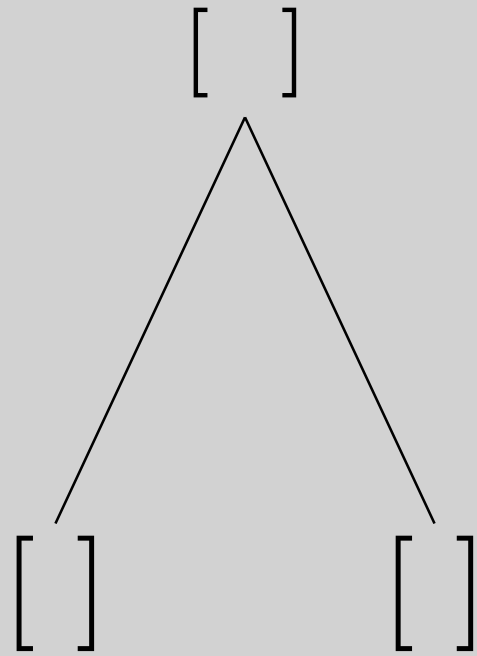
Effect of Principles: the SHAC



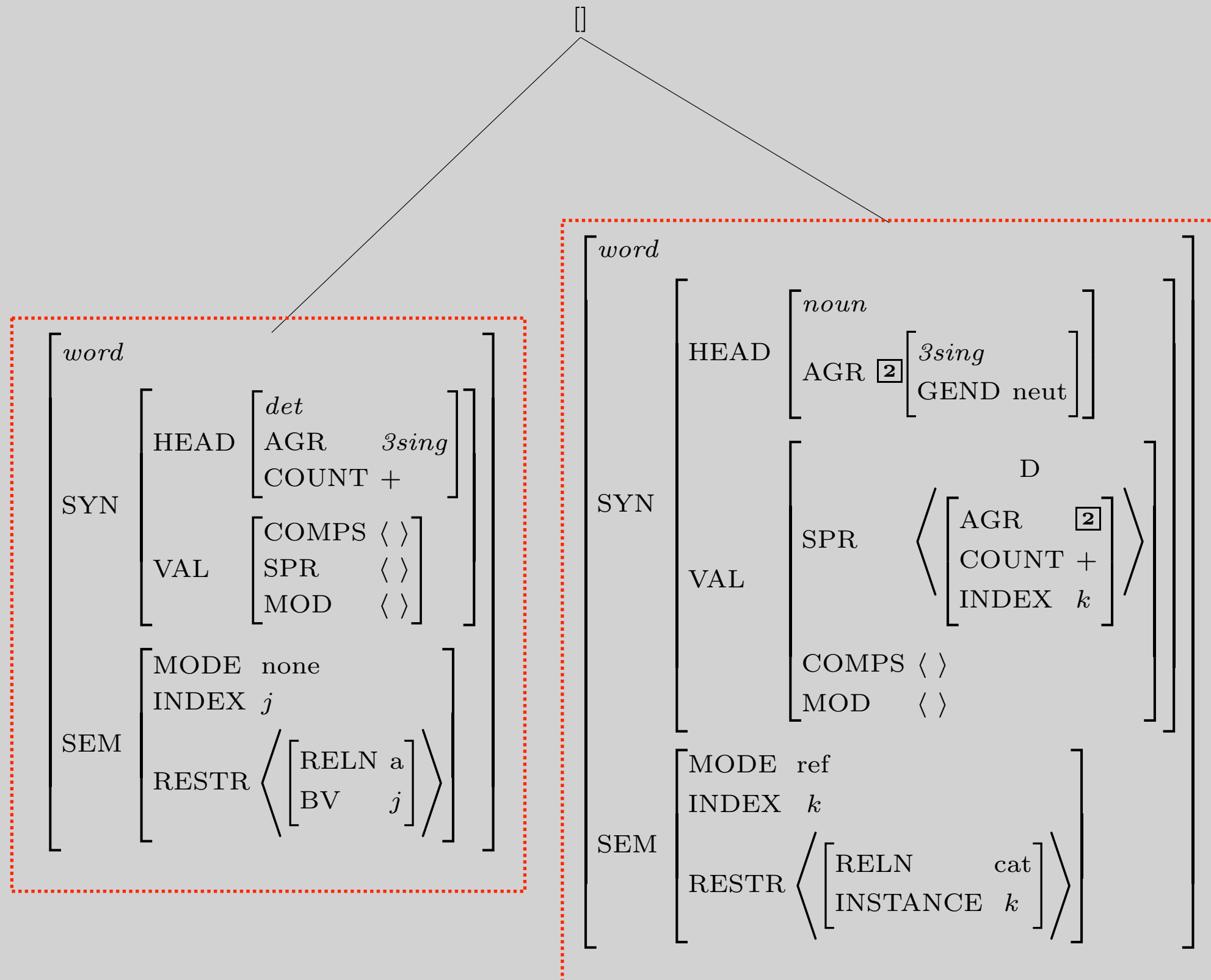
Description of Word Structures for *cat*



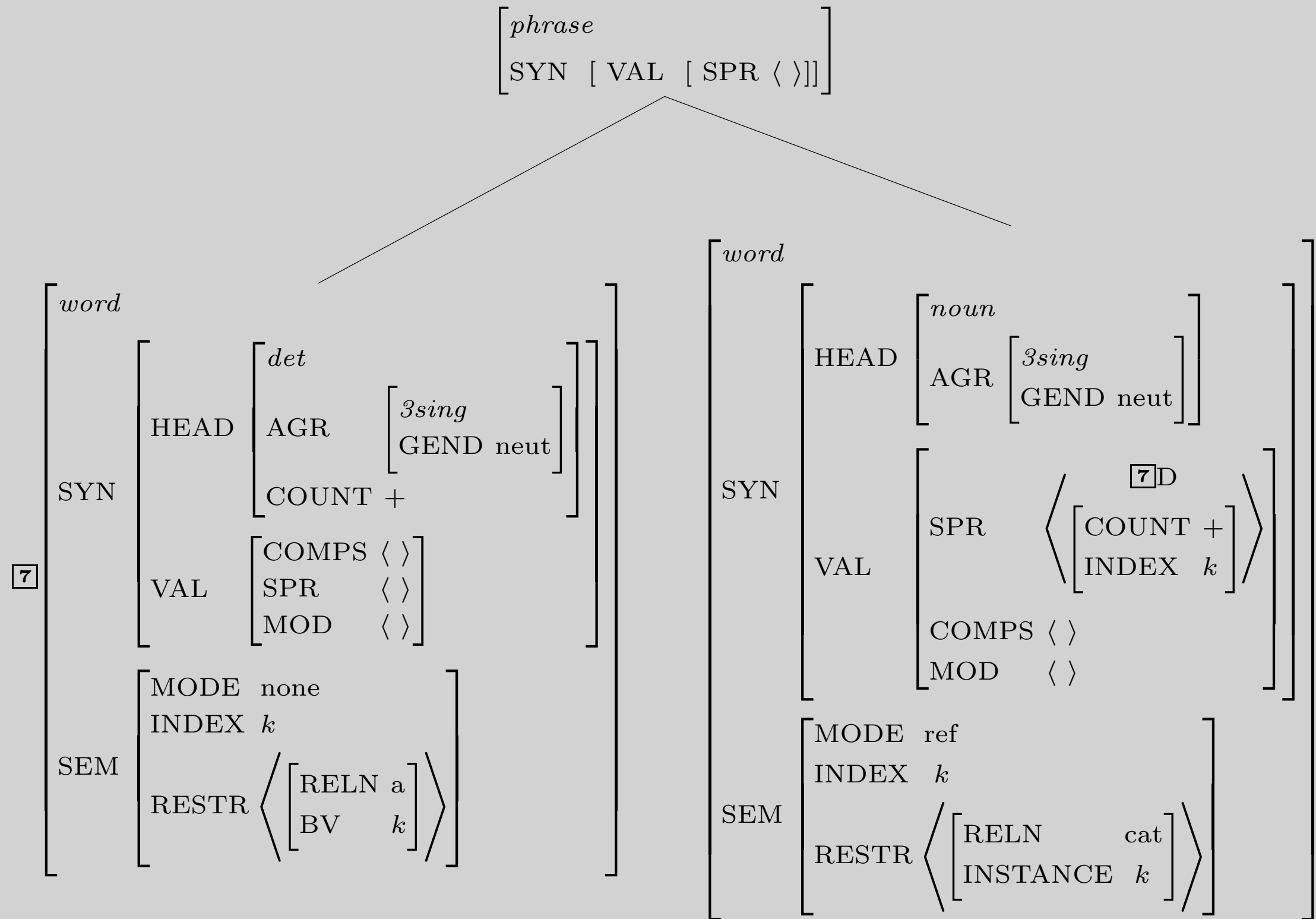
Building a Phrase



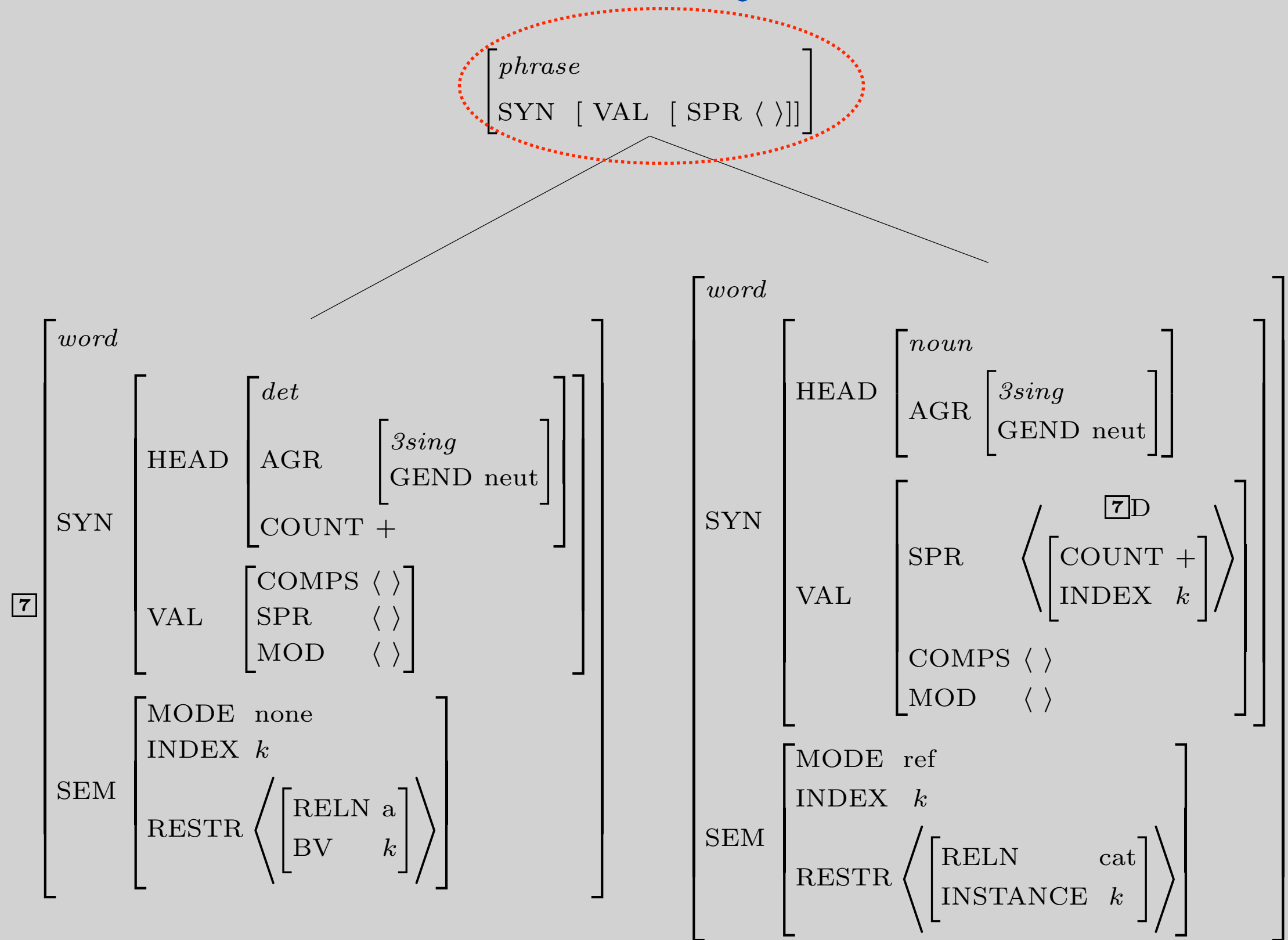
Constraints Contributed by Daughter Subtrees



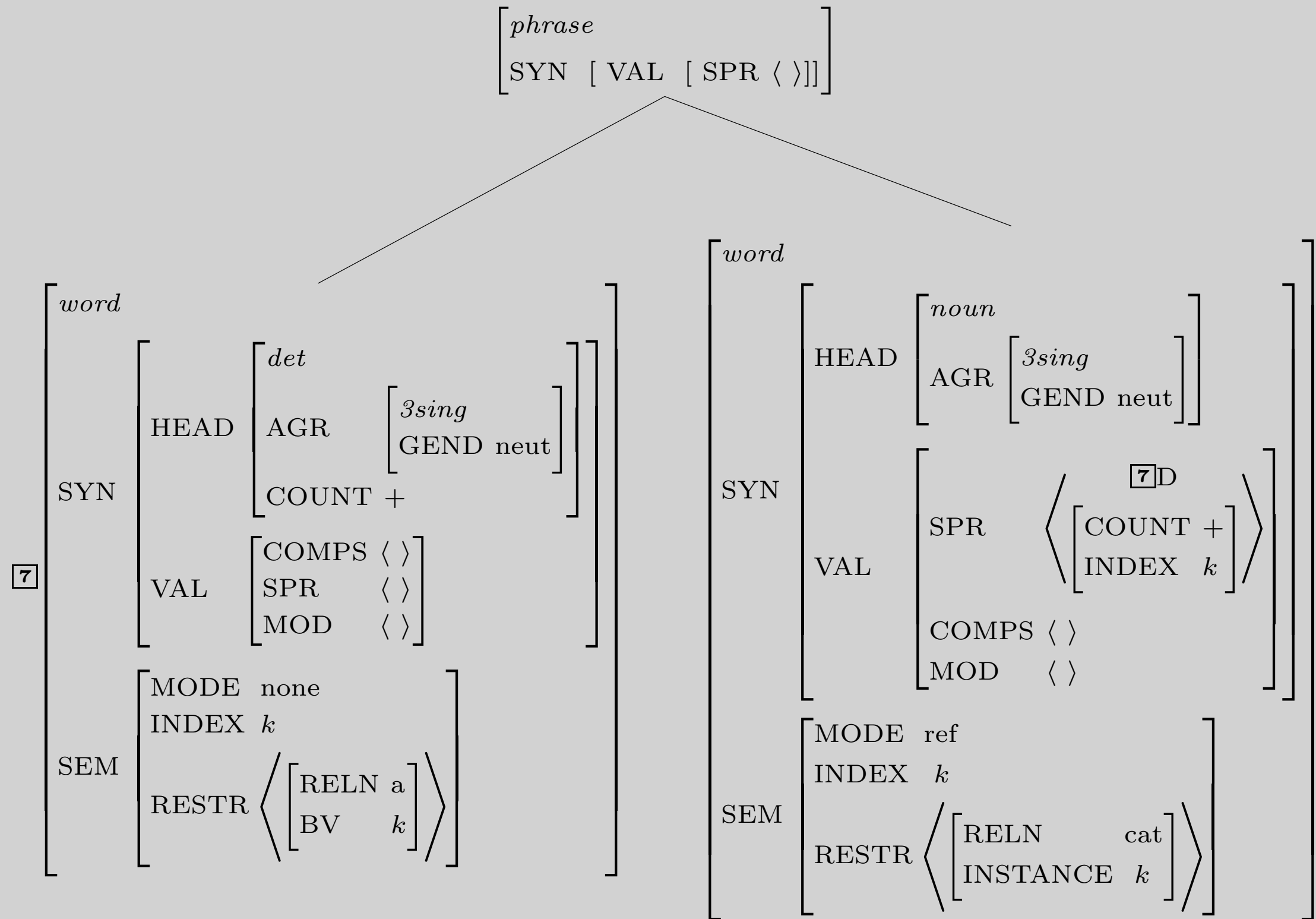
Constraints Contributed by the Grammar Rule



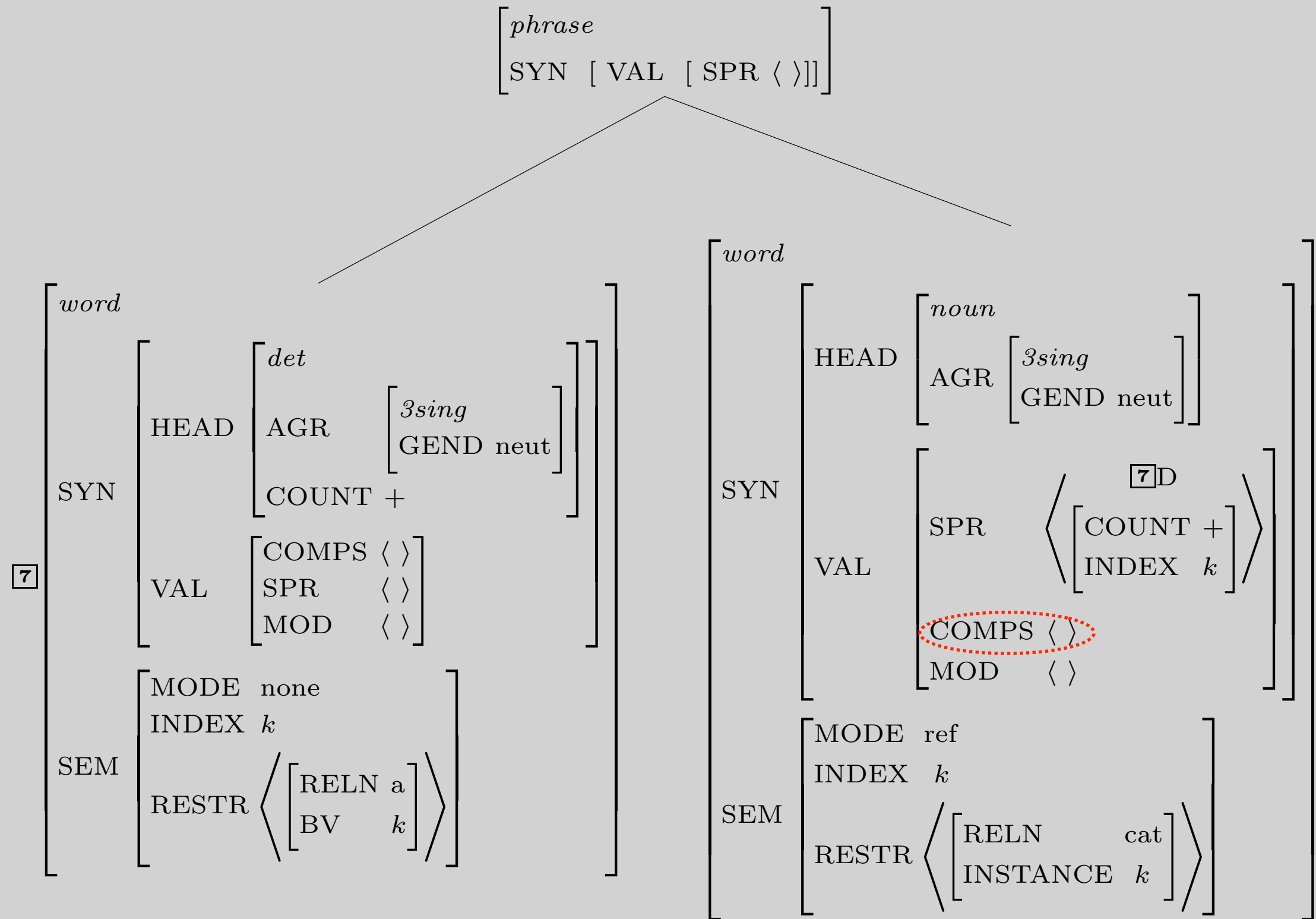
Constraints Contributed by the Grammar Rule



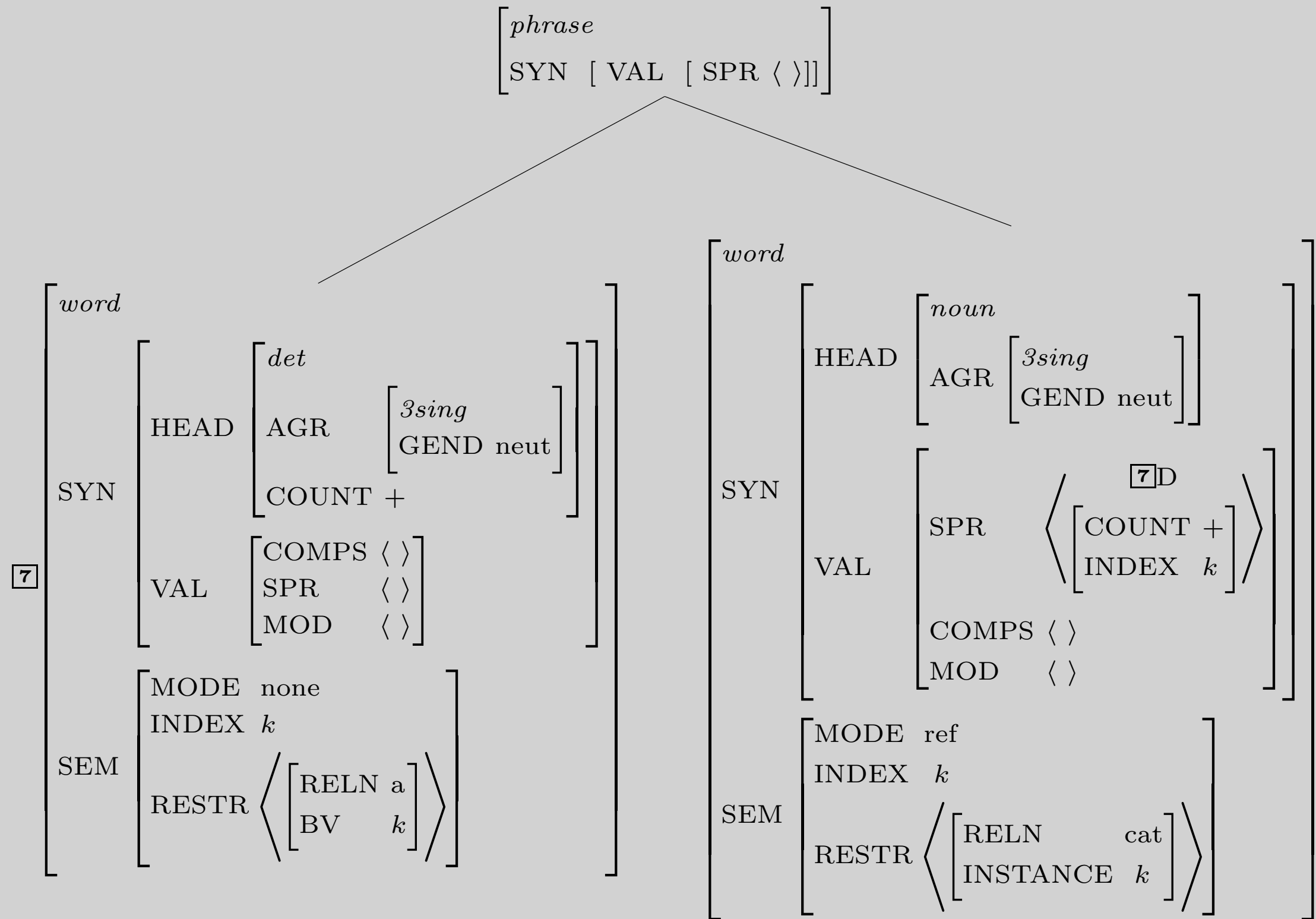
Constraints Contributed by the Grammar Rule



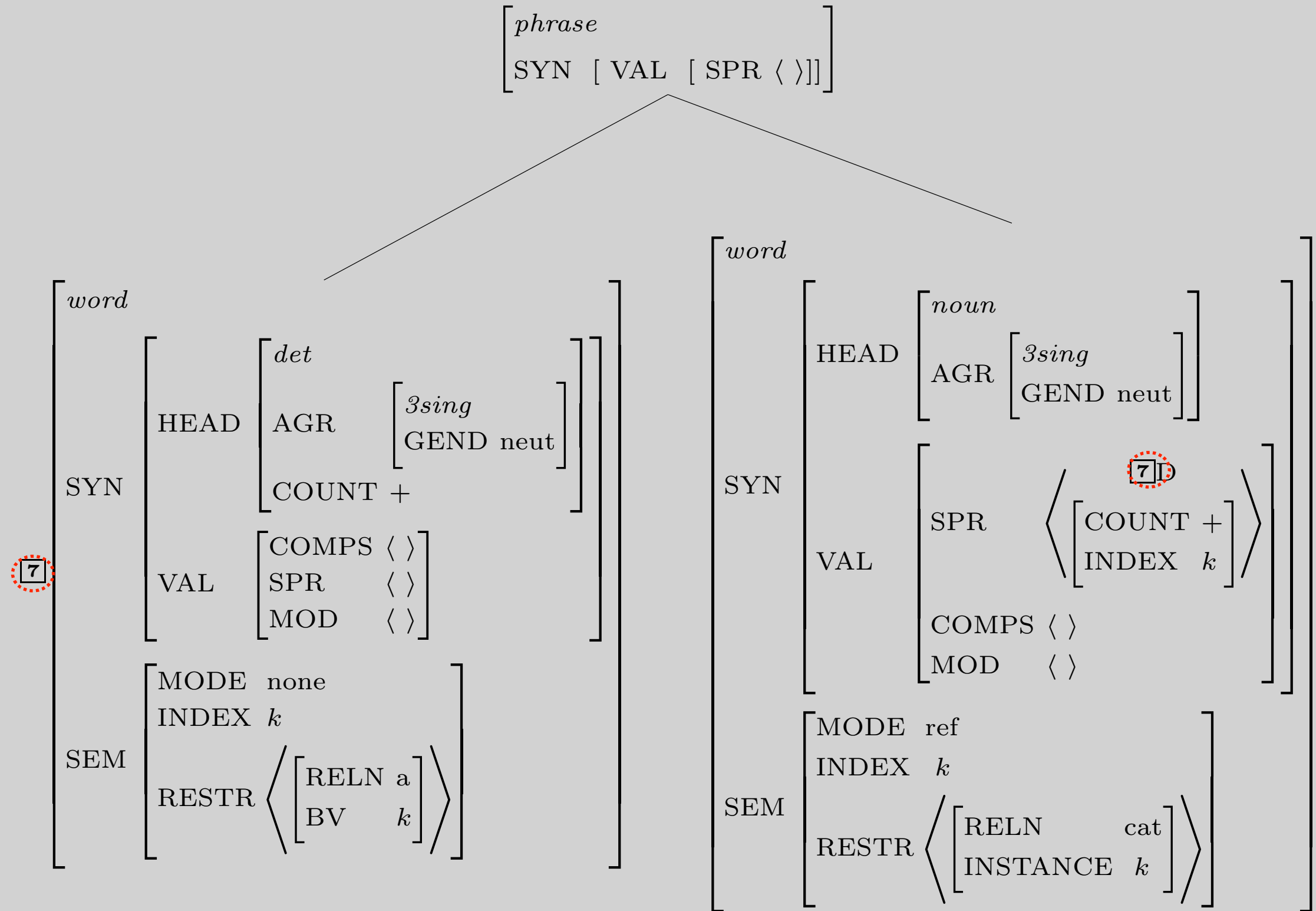
Constraints Contributed by the Grammar Rule



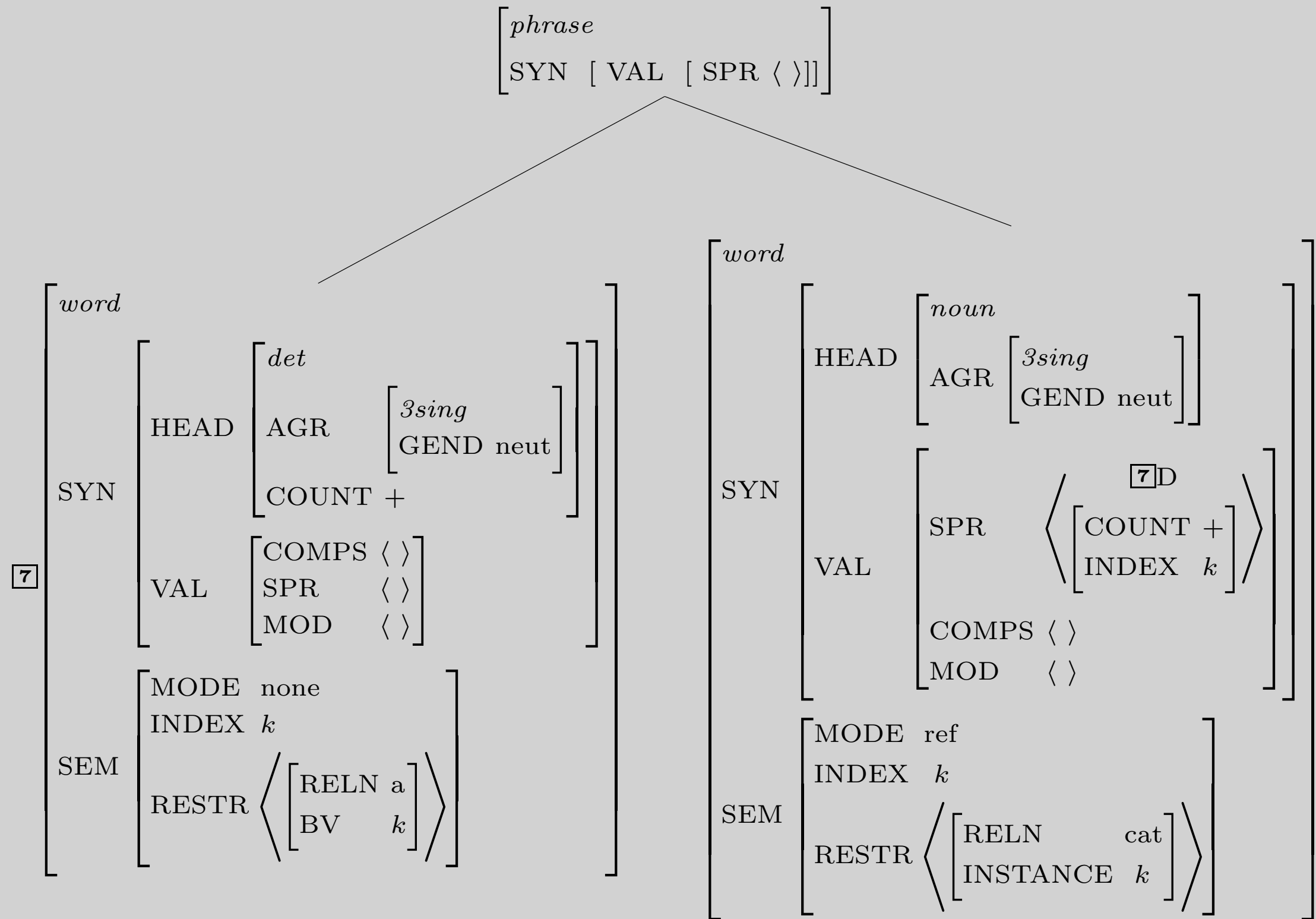
Constraints Contributed by the Grammar Rule



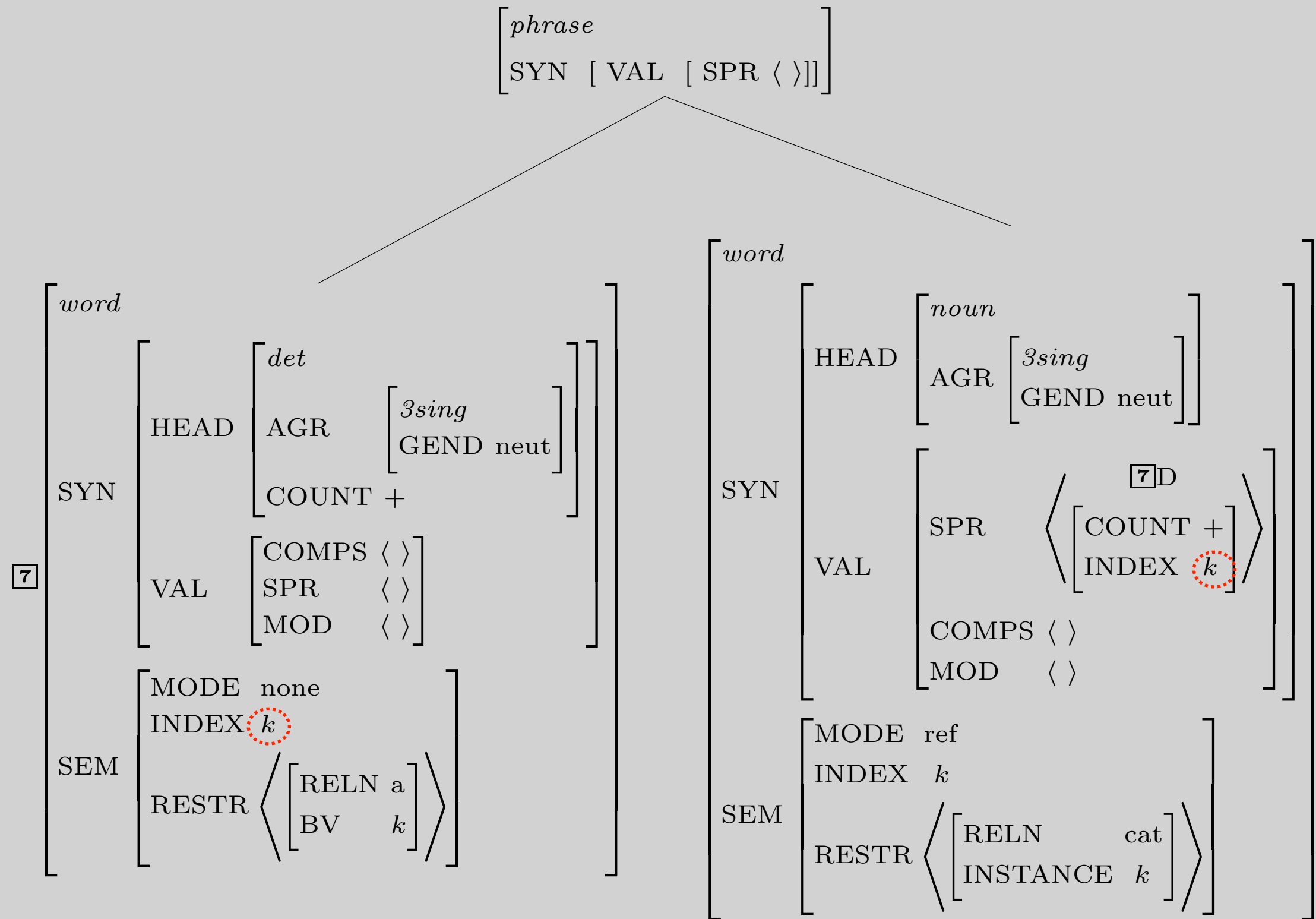
Constraints Contributed by the Grammar Rule



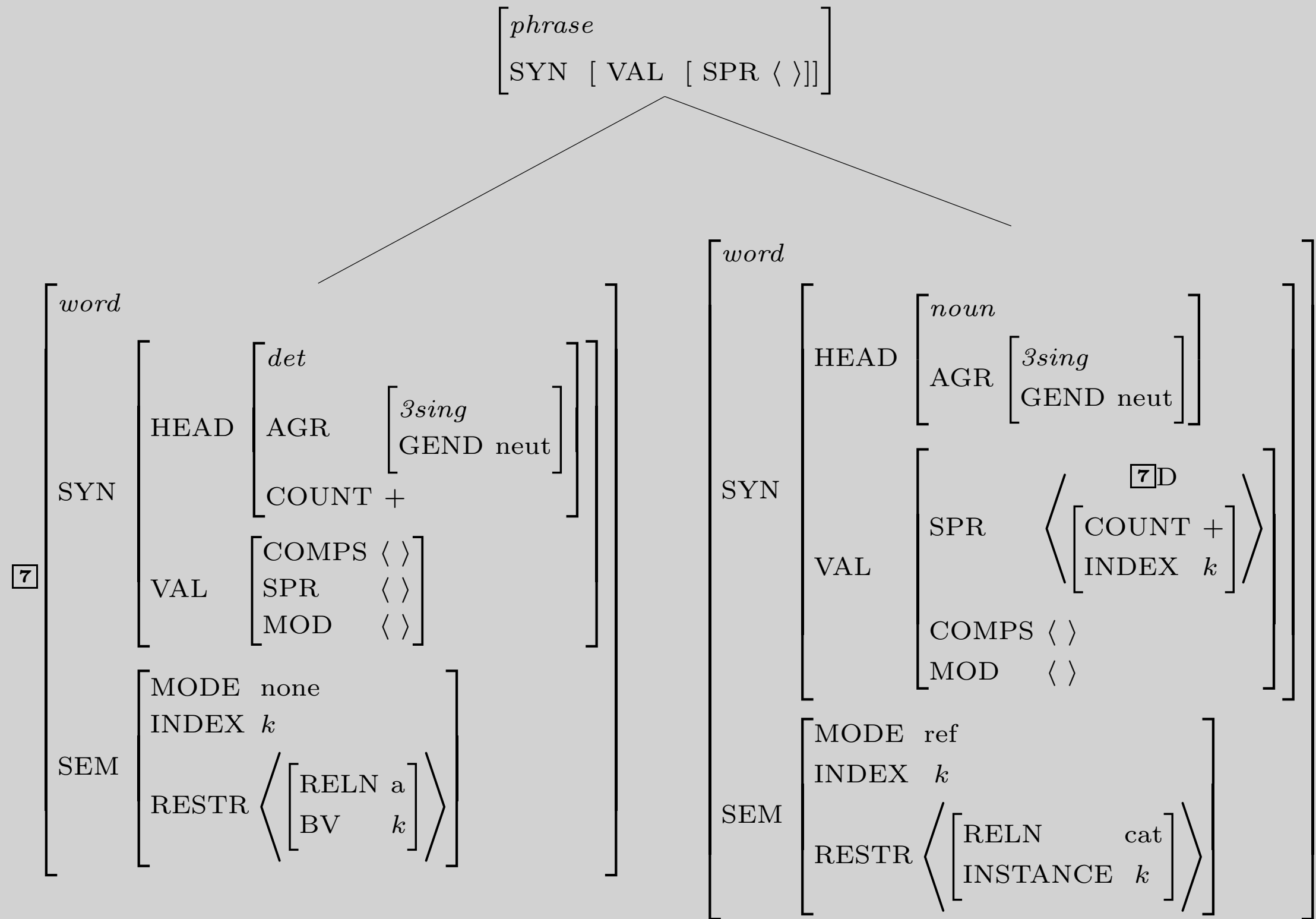
Constraints Contributed by the Grammar Rule



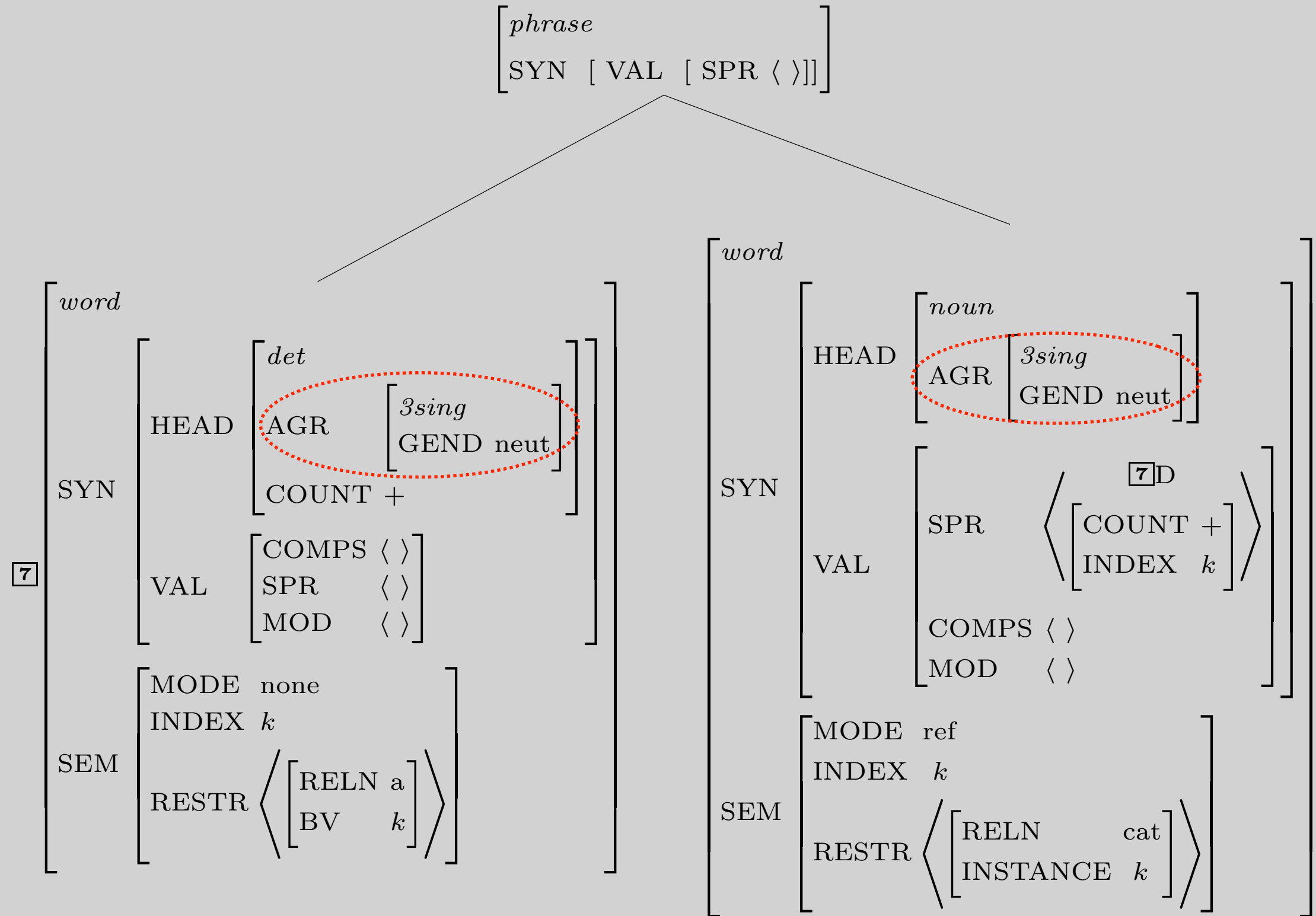
Constraints Contributed by the Grammar Rule



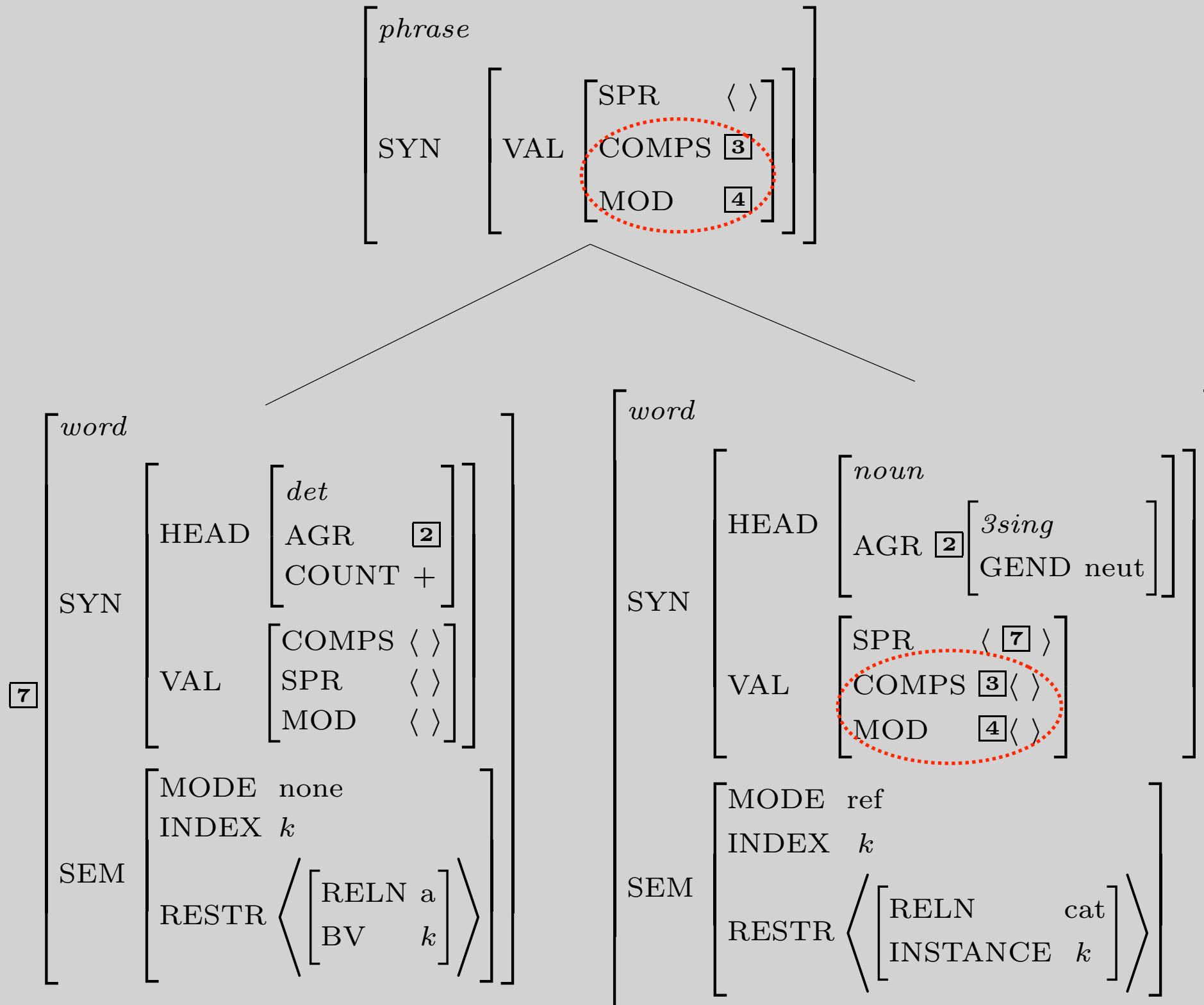
Constraints Contributed by the Grammar Rule



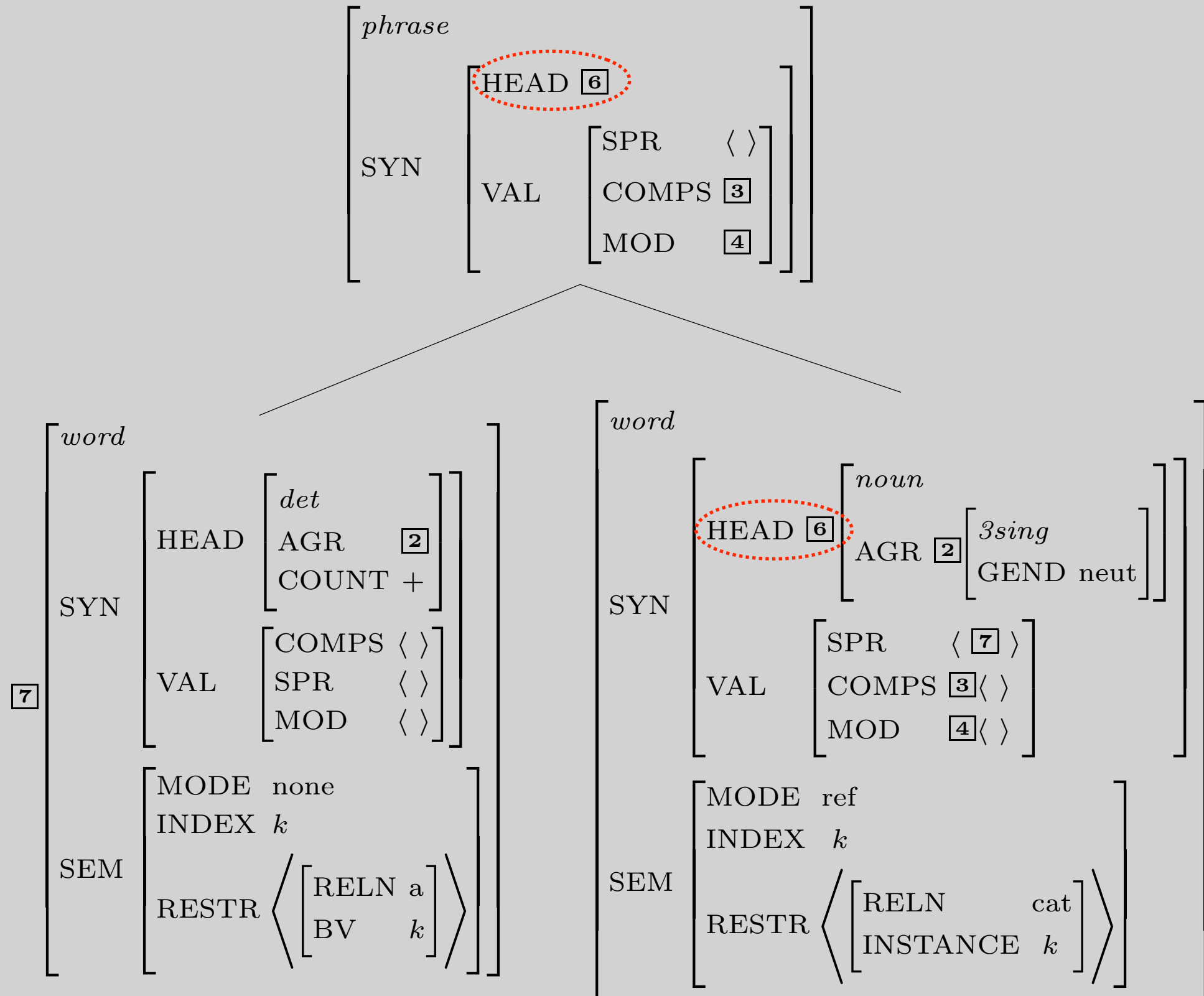
A Constraint Involving the SHAC



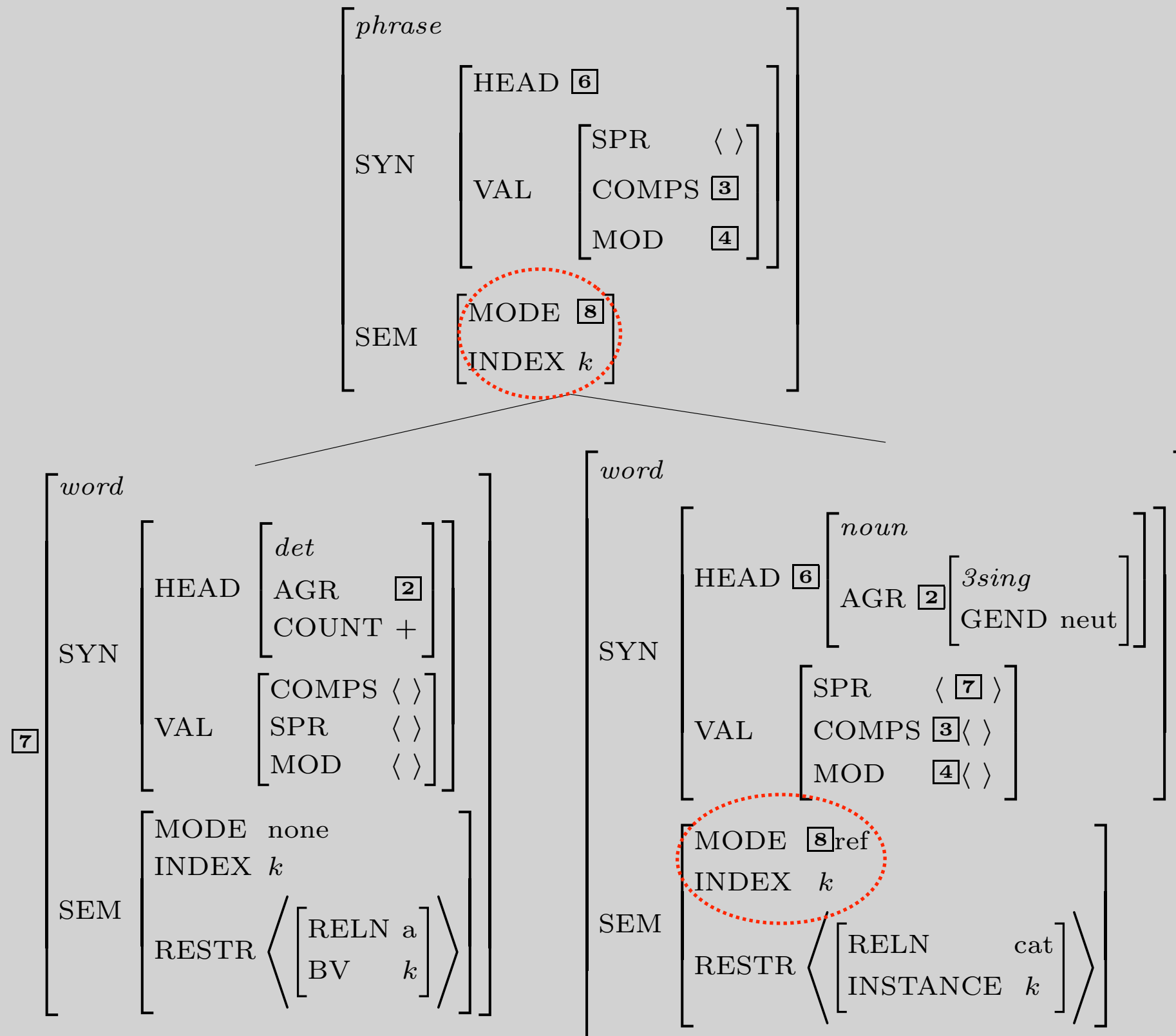
Effects of the Valence Principle



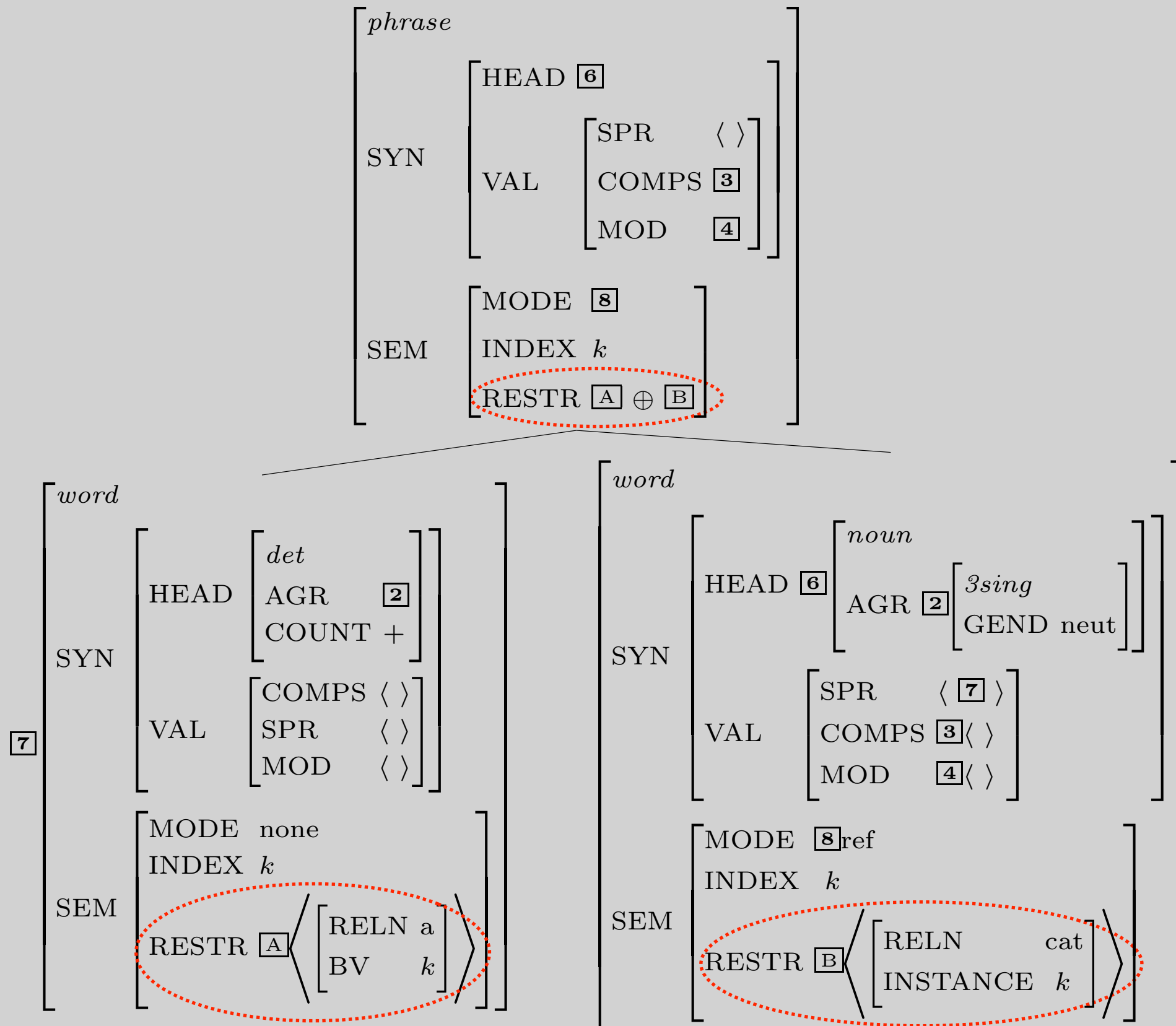
Effects of the Head Feature Principle



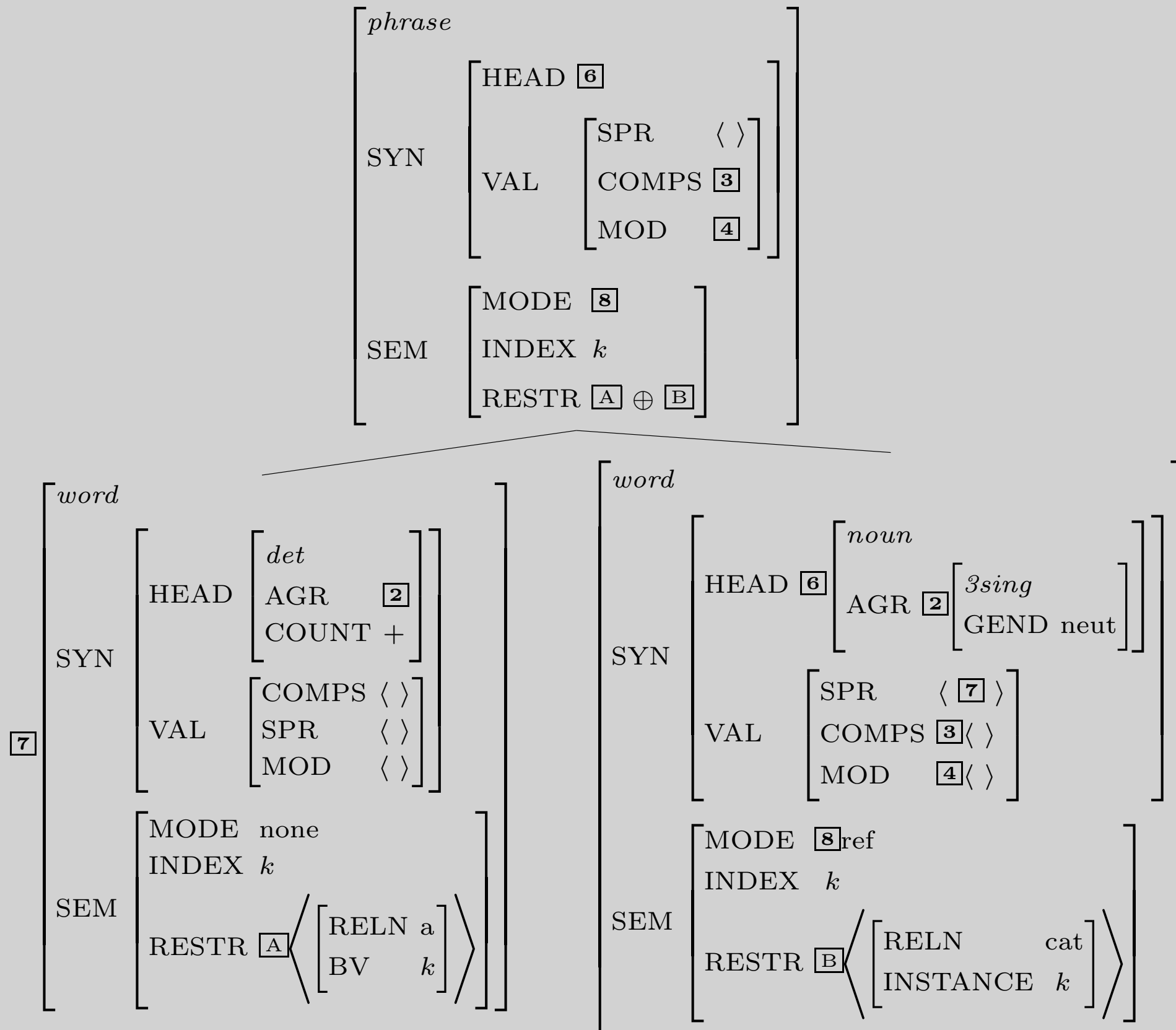
Effects of the Semantic Inheritance Principle



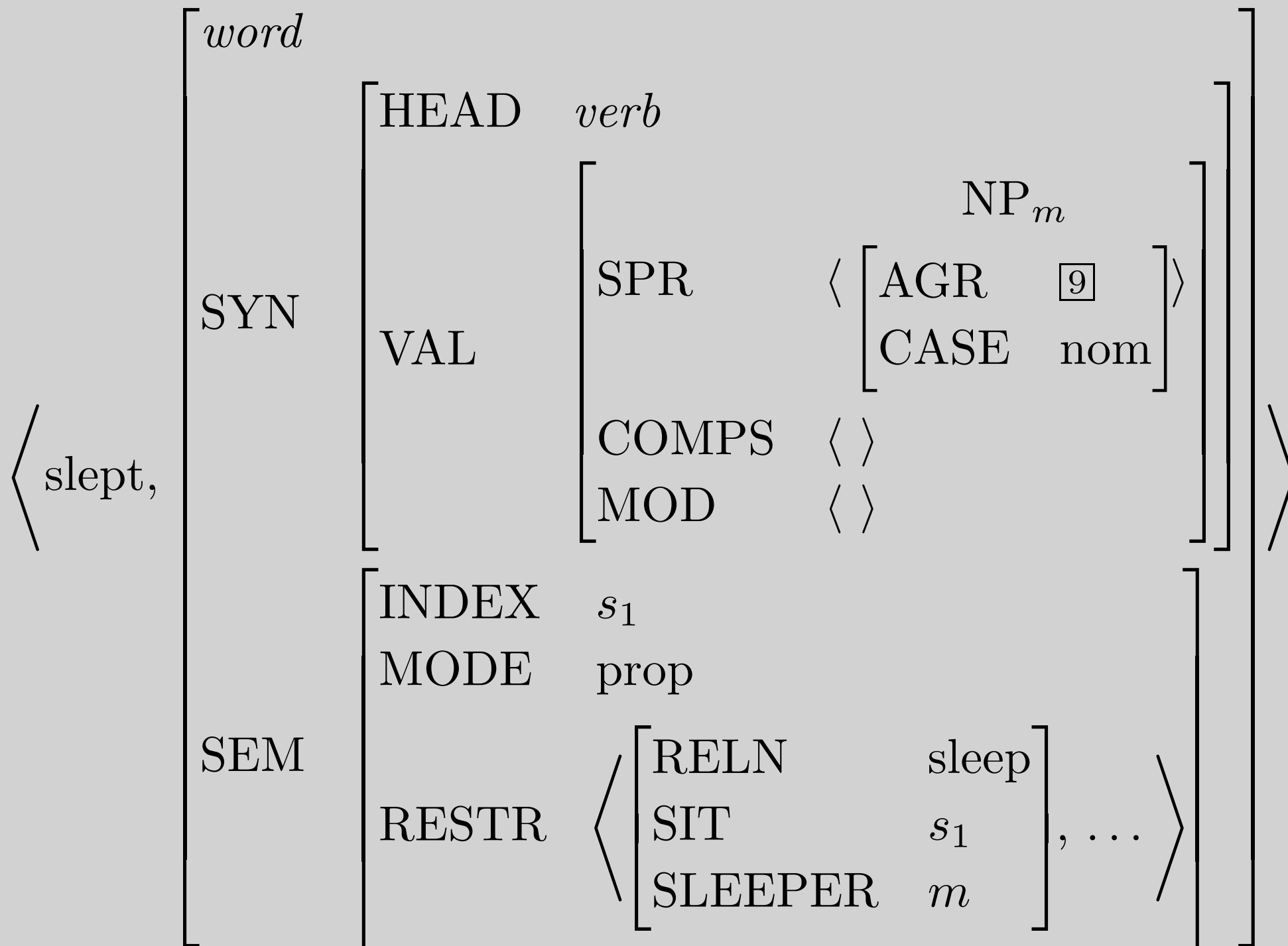
Effects of the Semantic Compositionality Principle



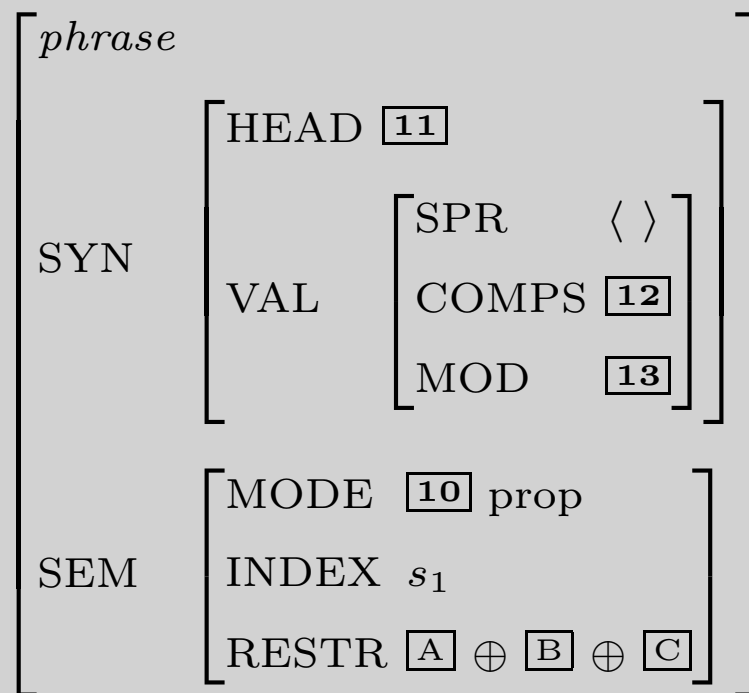
Is the Mother Node Now Completely Specified?



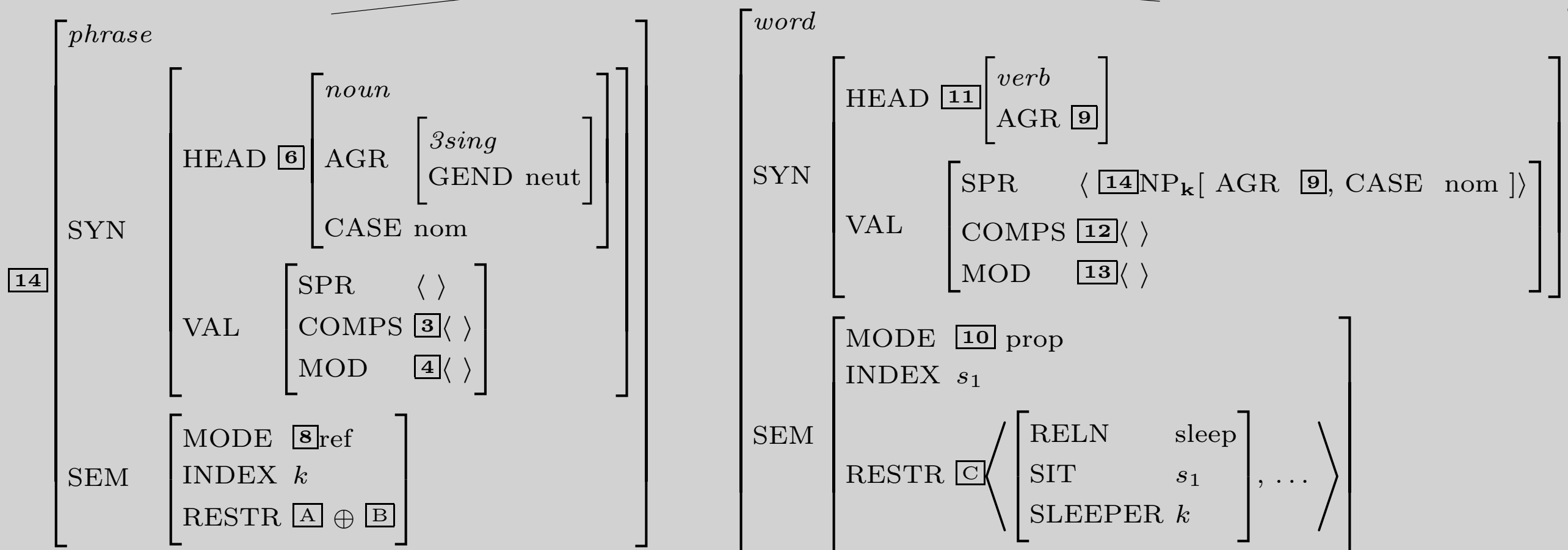
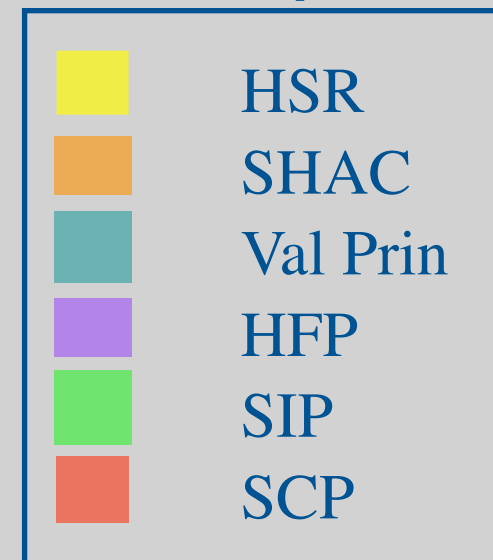
Lexical Entry for *slept*



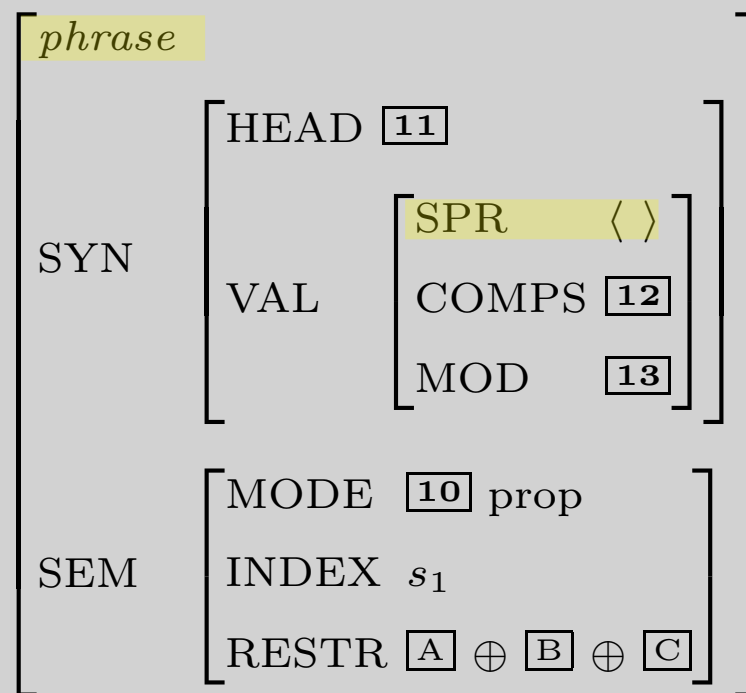
Another Head-Specifier Phrase



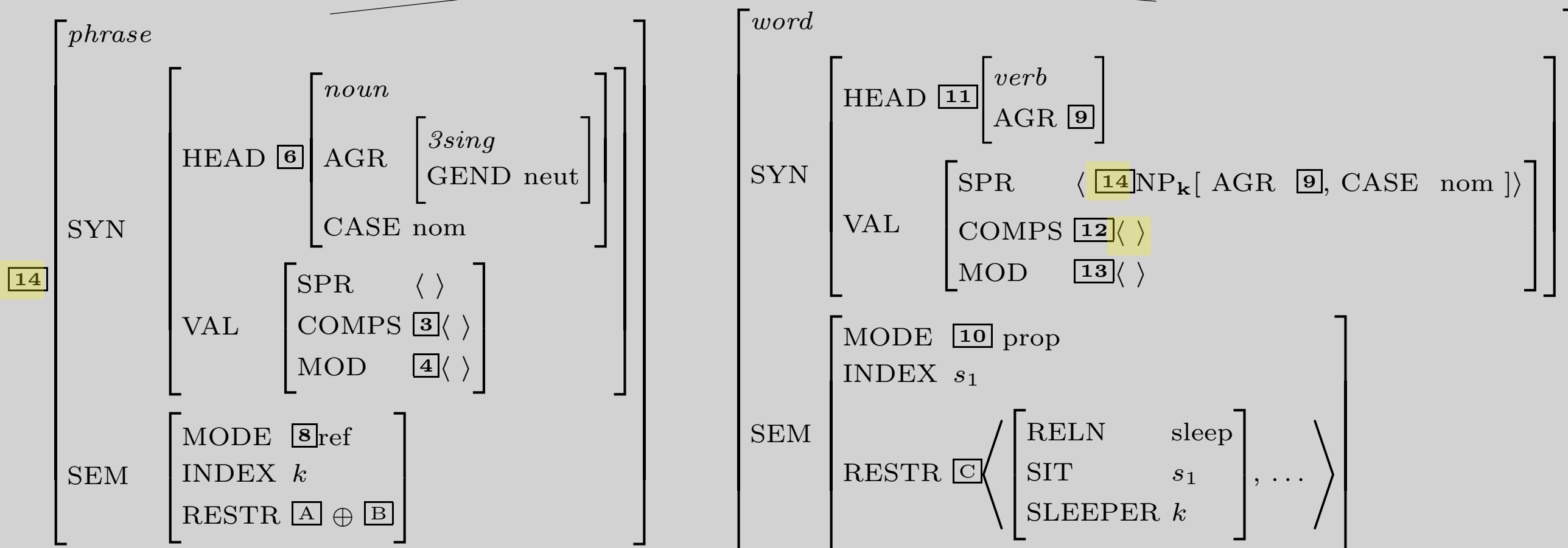
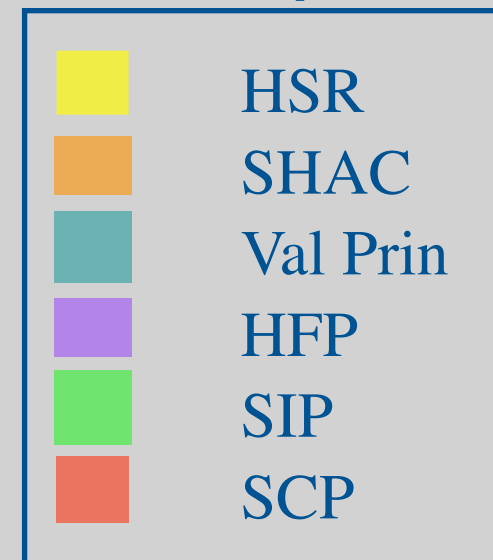
Key



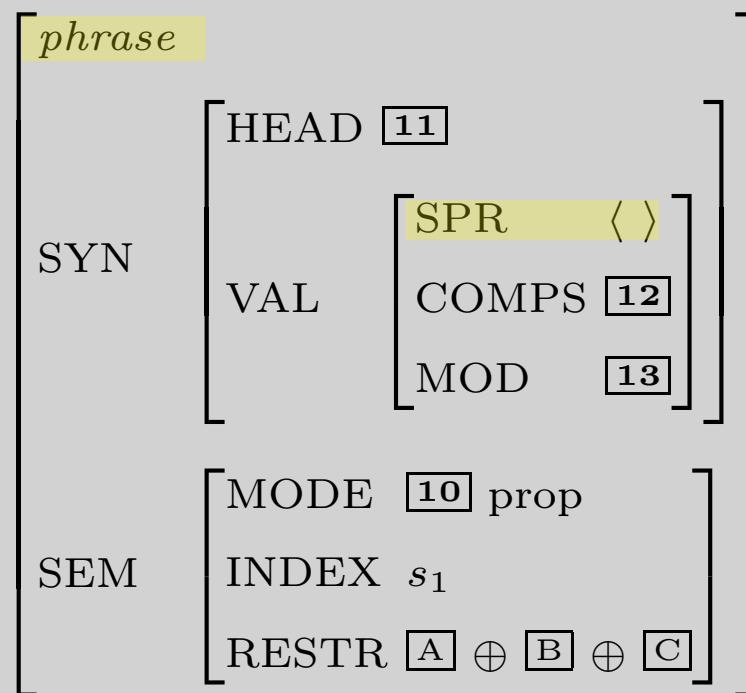
Another Head-Specifier Phrase



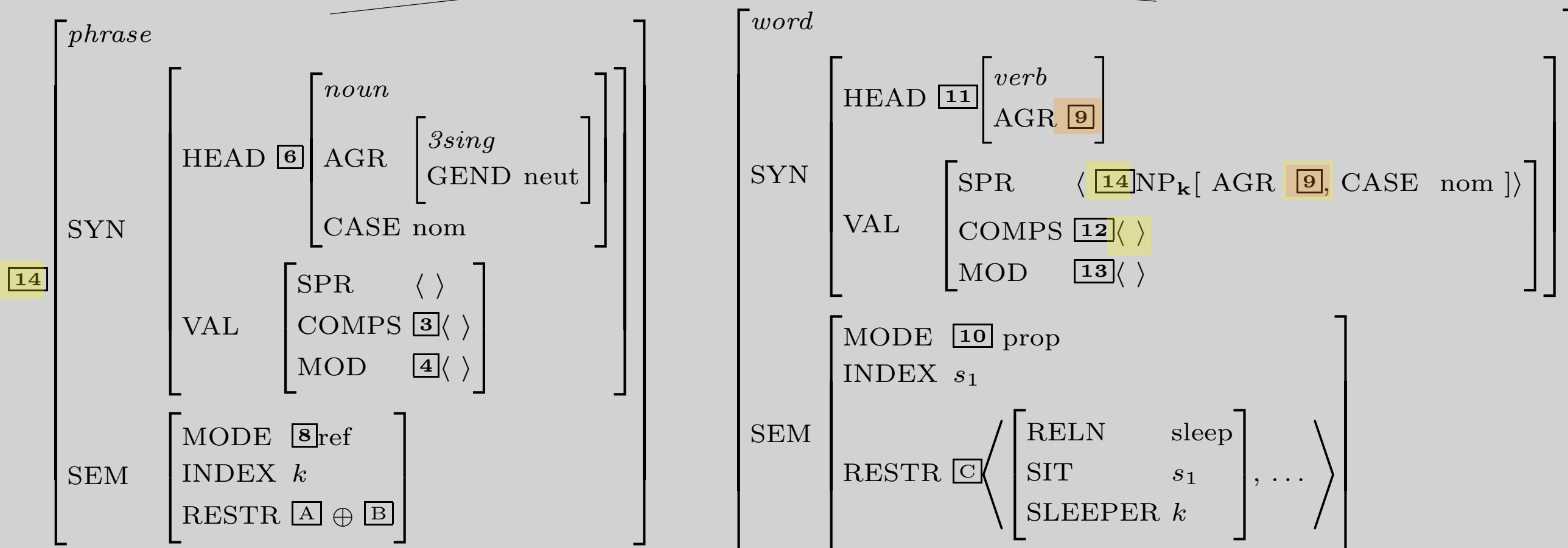
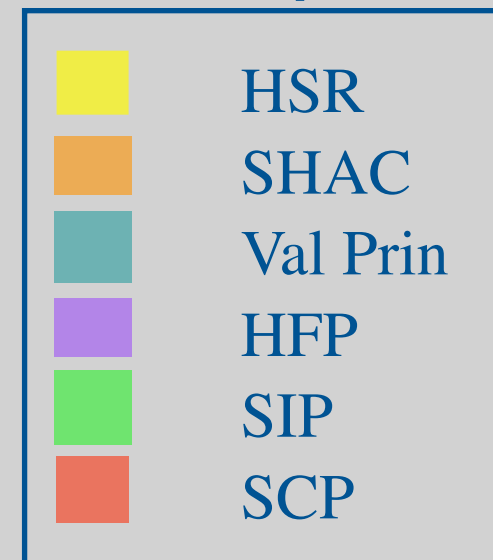
Key



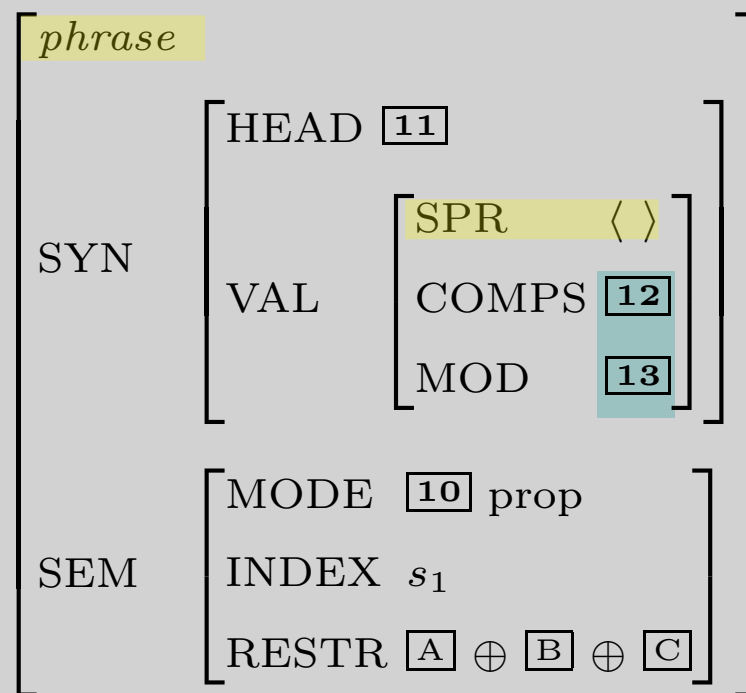
Another Head-Specifier Phrase



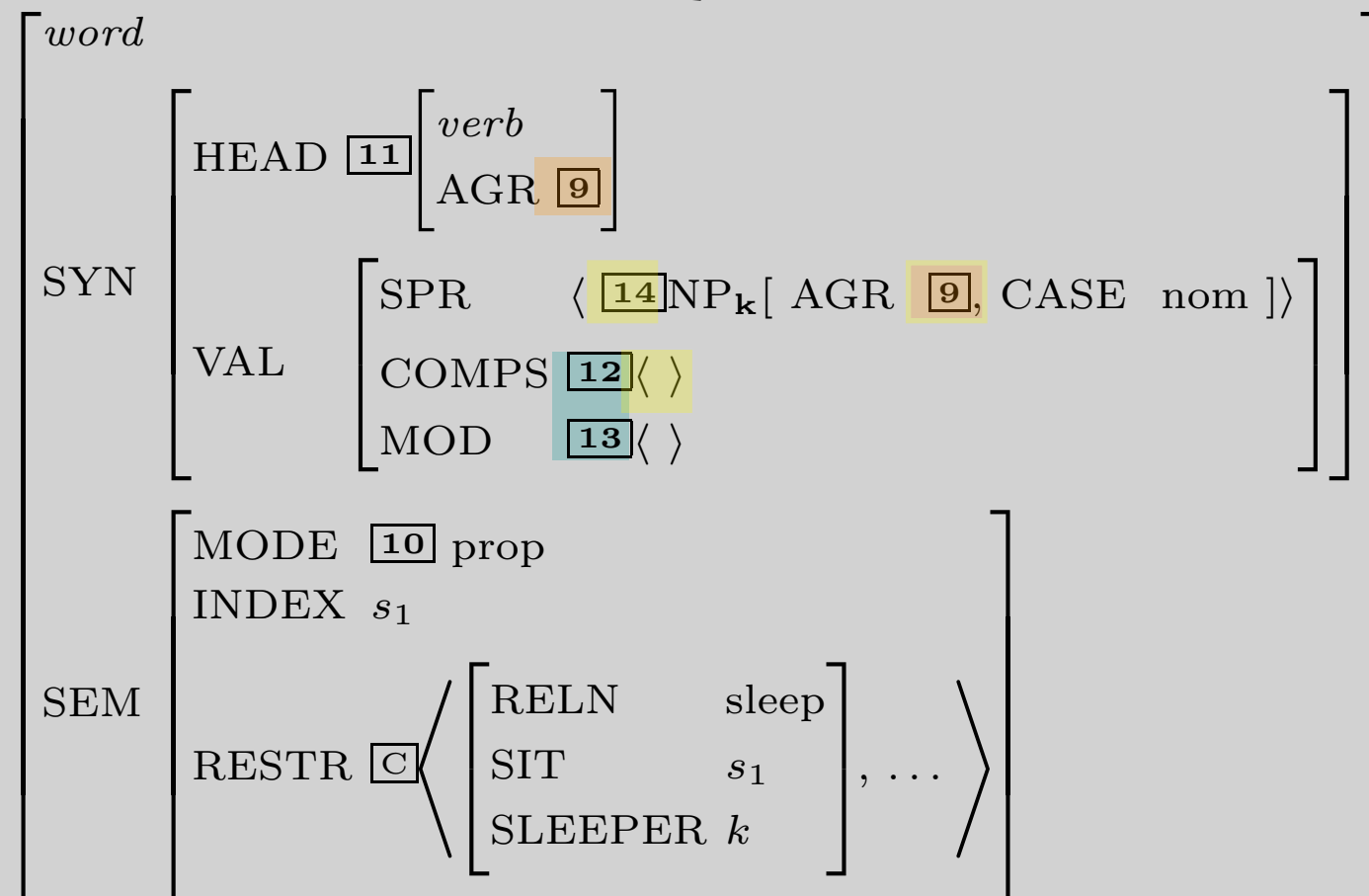
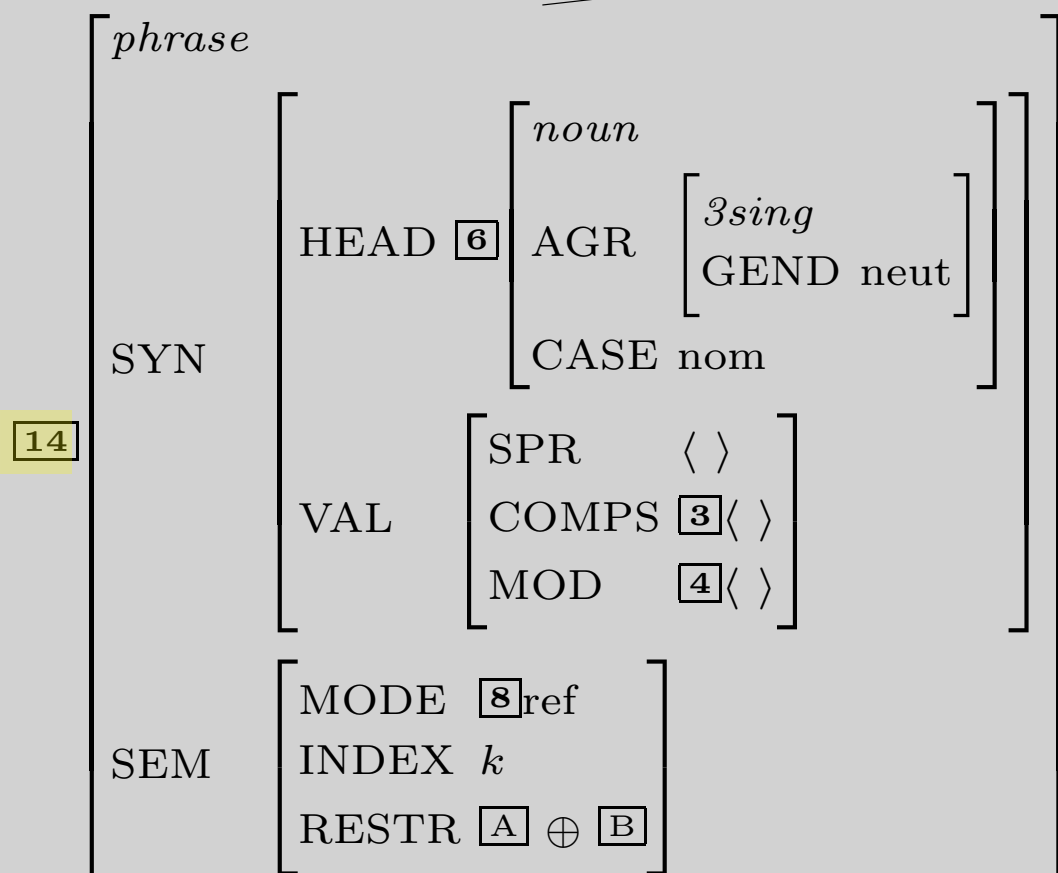
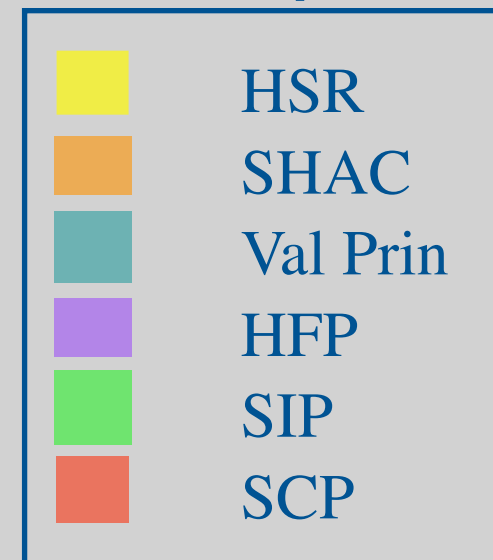
Key



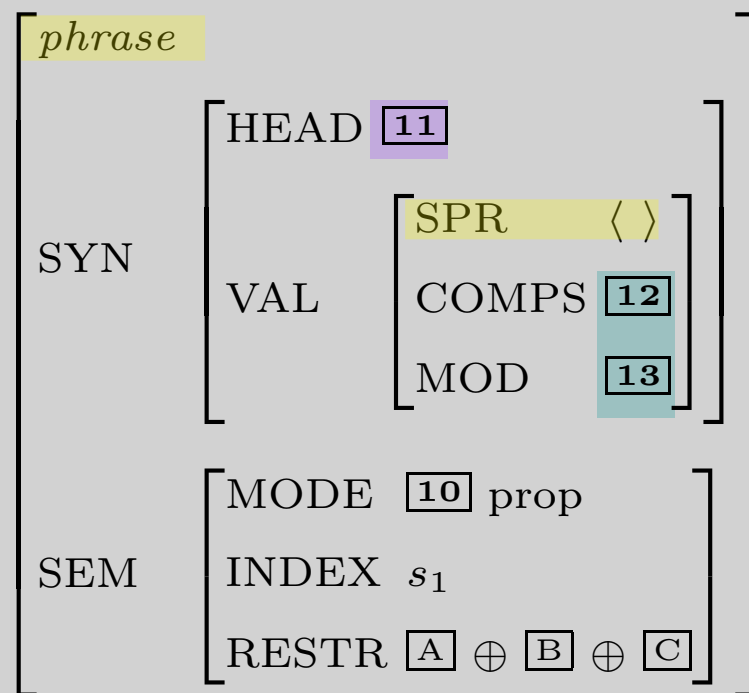
Another Head-Specifier Phrase



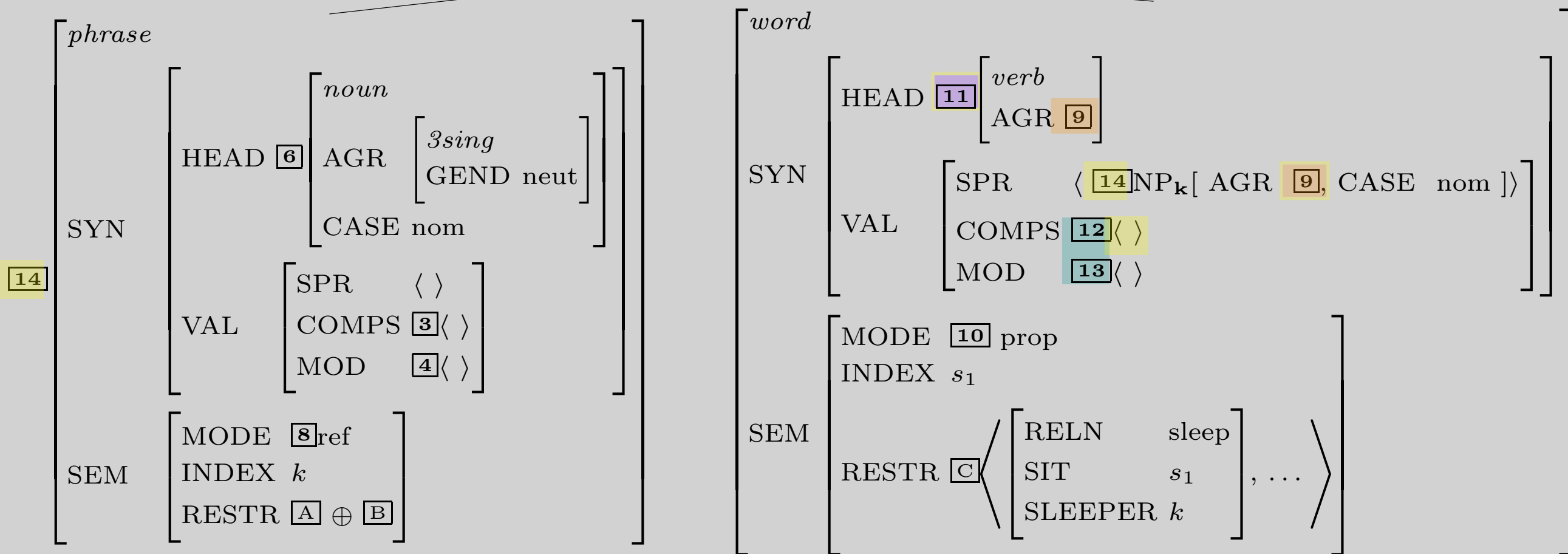
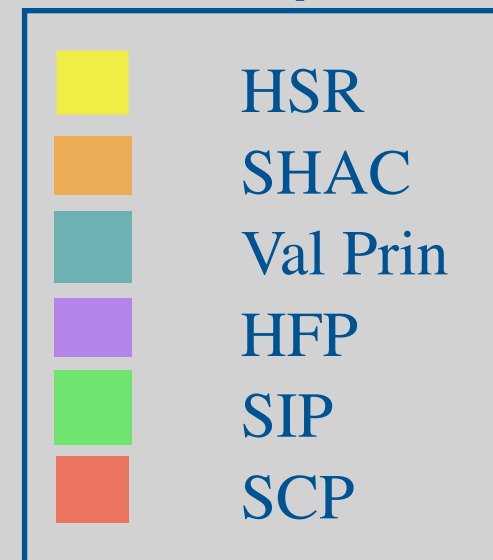
Key



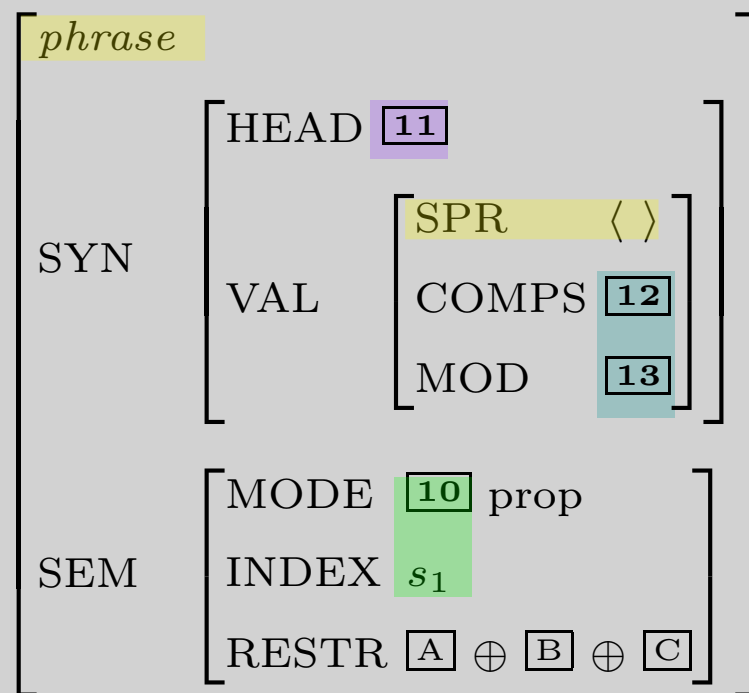
Another Head-Specifier Phrase



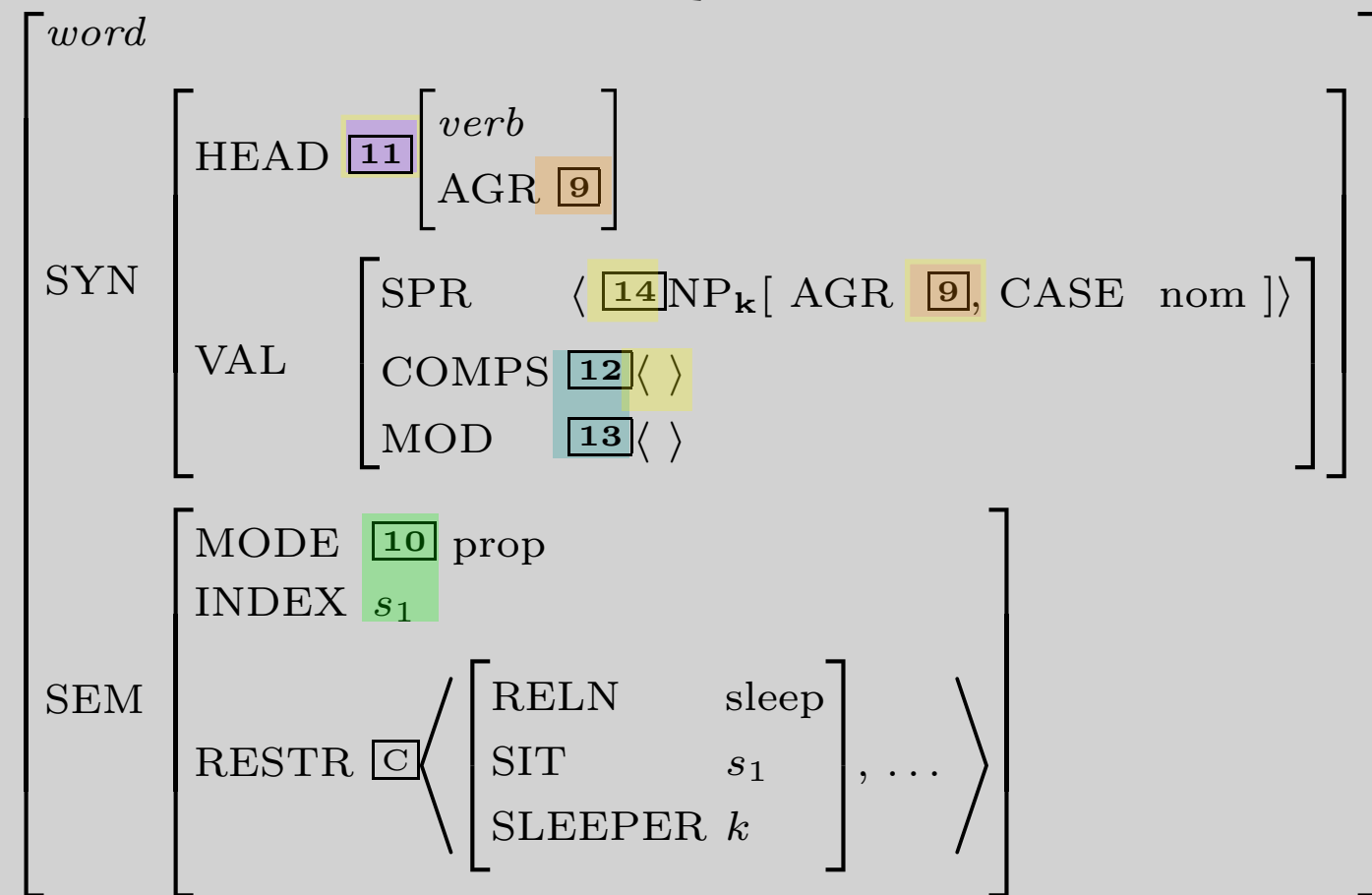
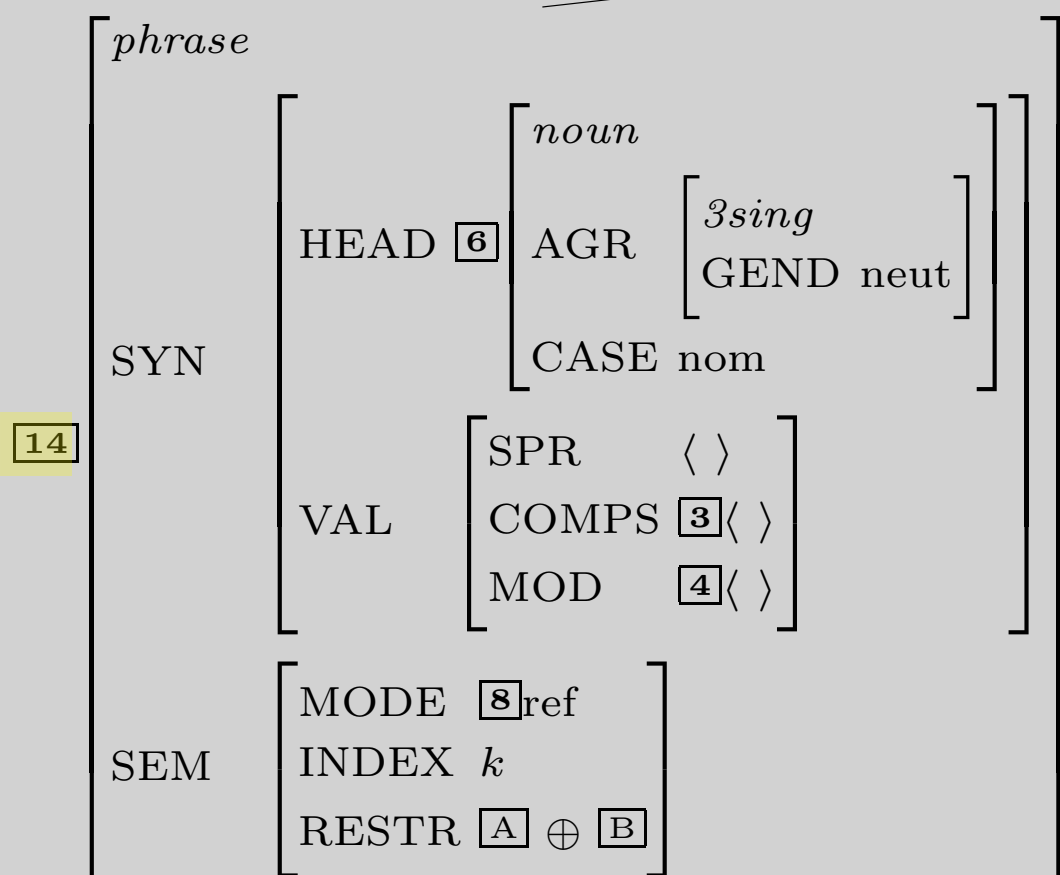
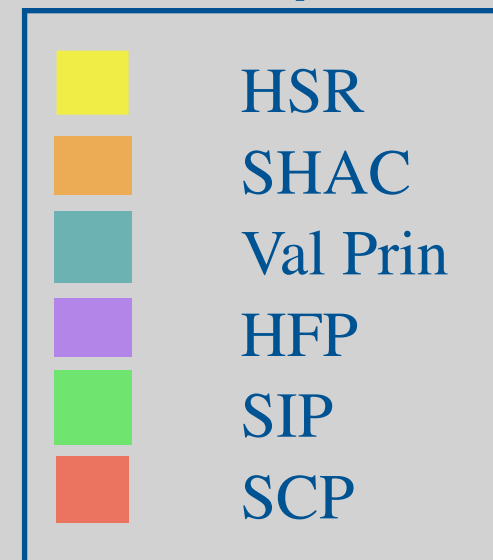
Key



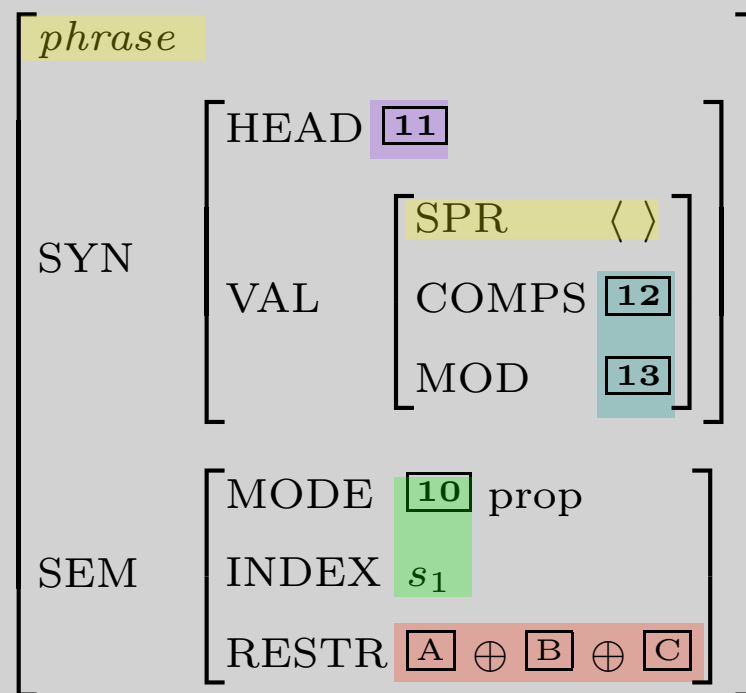
Another Head-Specifier Phrase



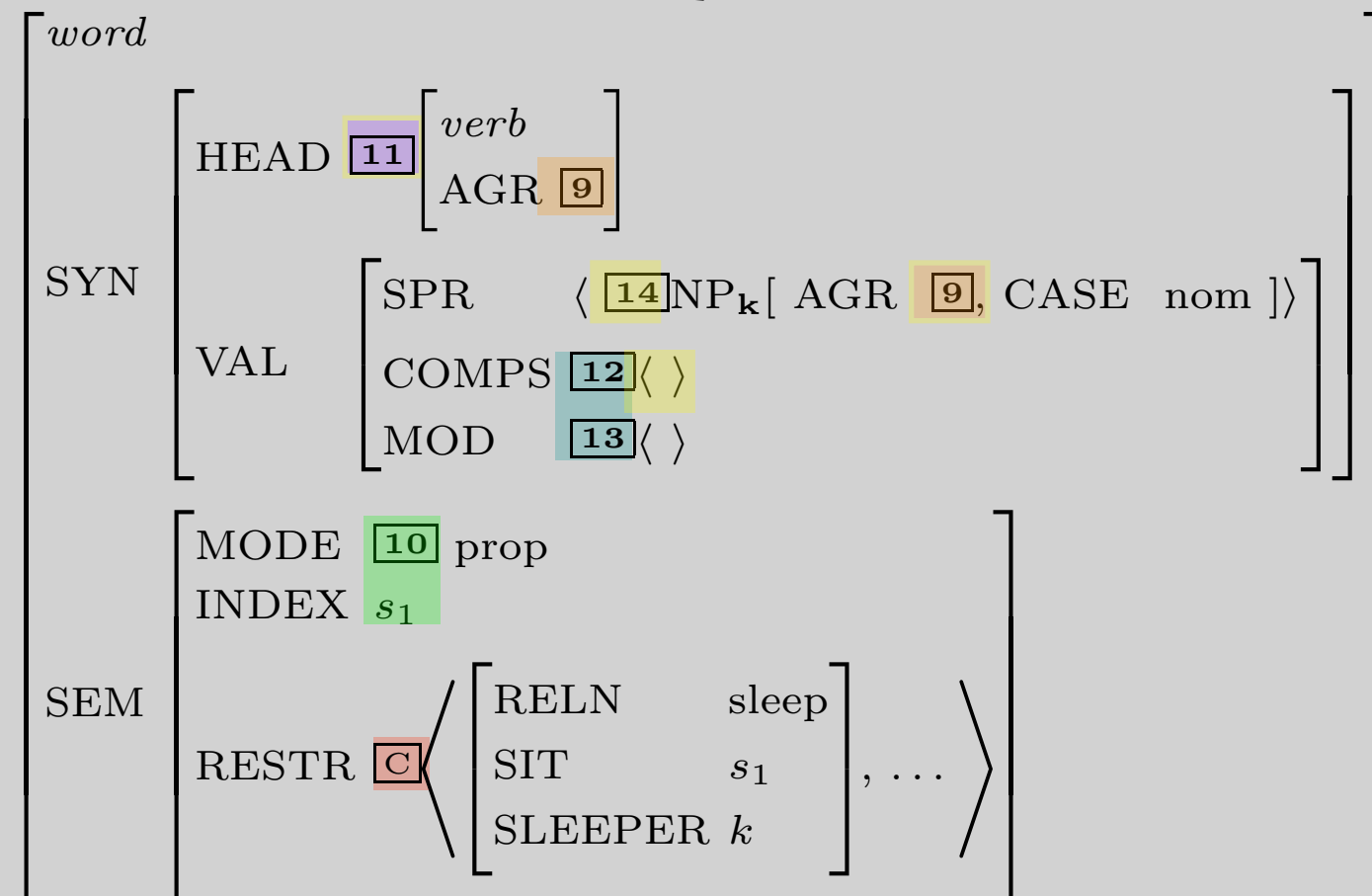
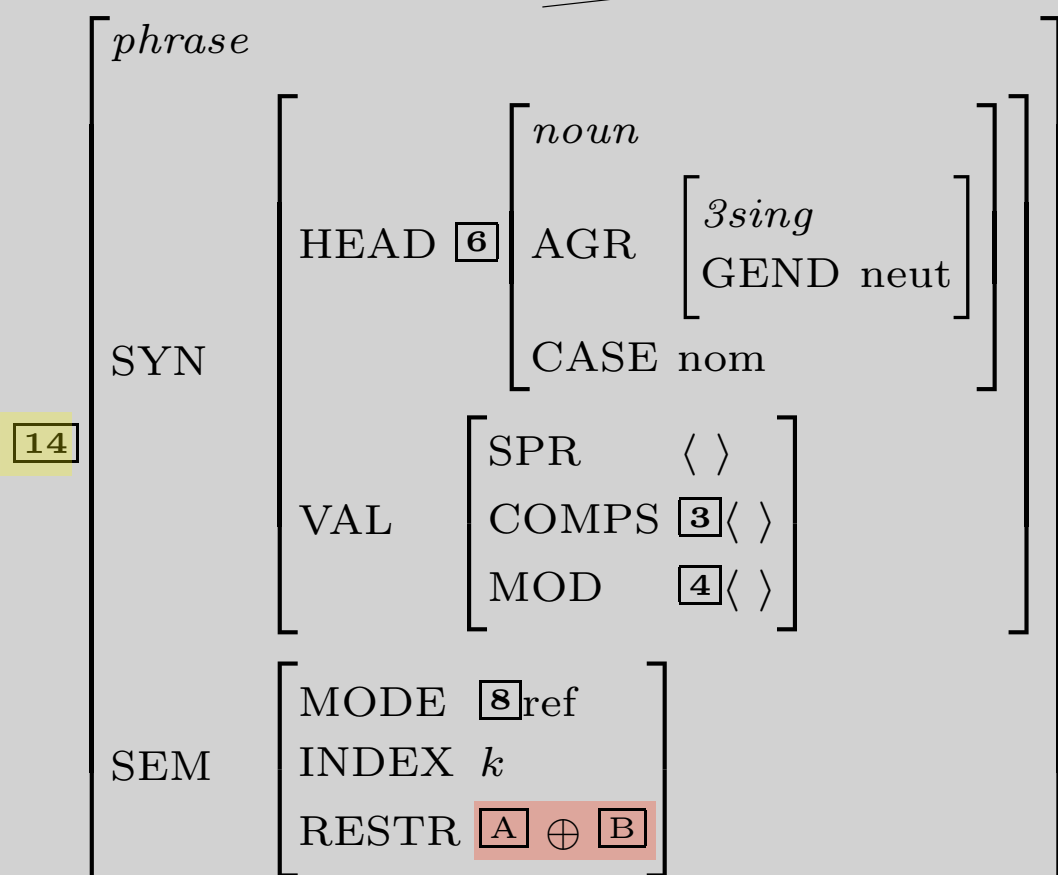
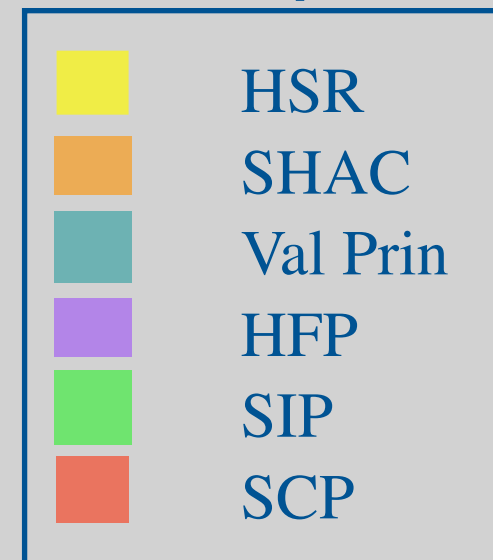
Key



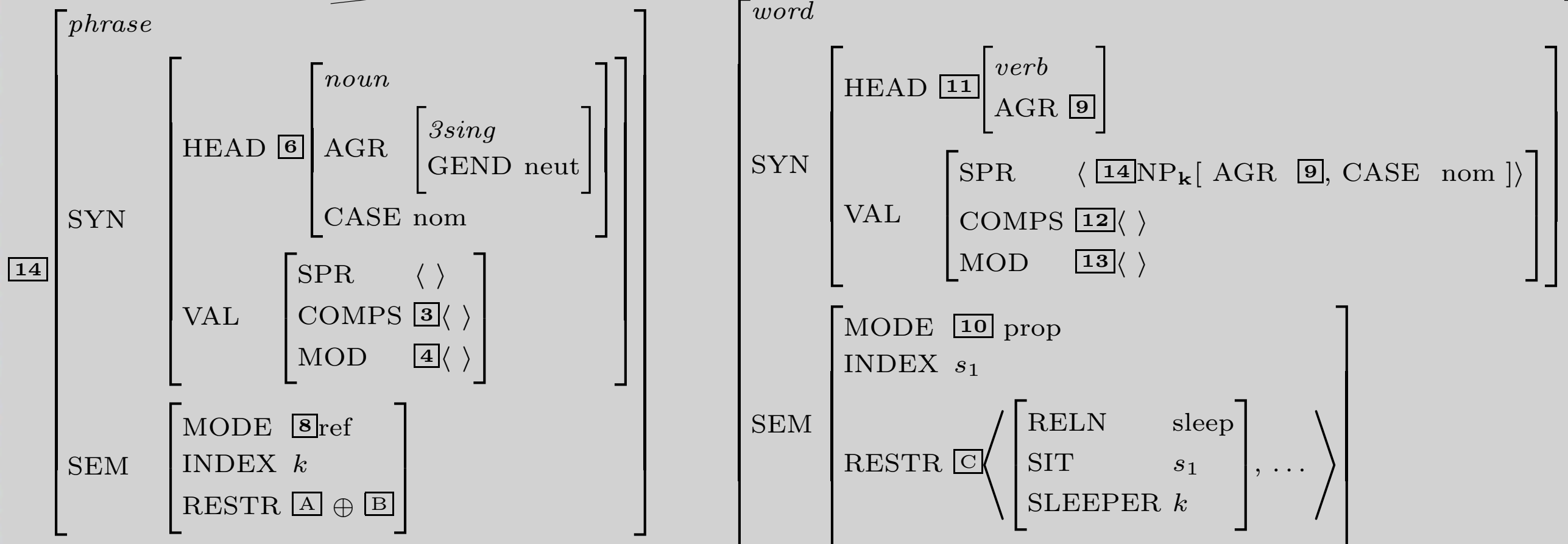
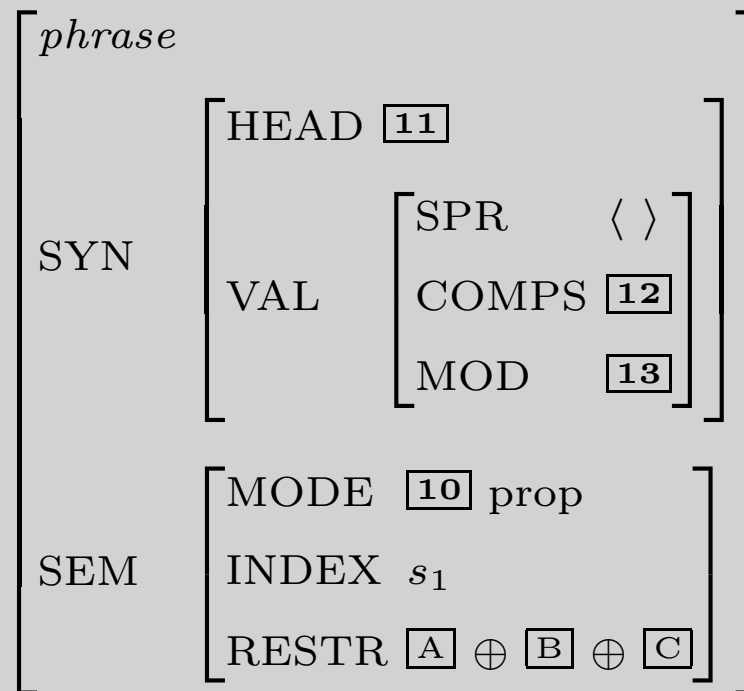
Another Head-Specifier Phrase



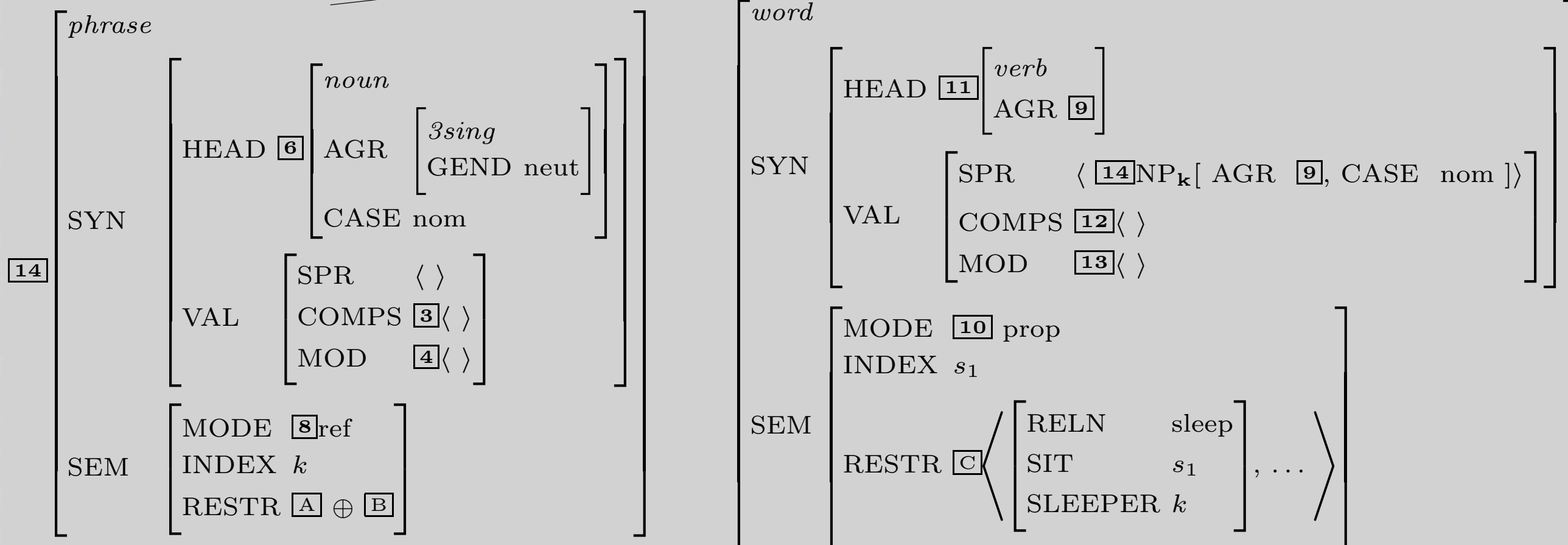
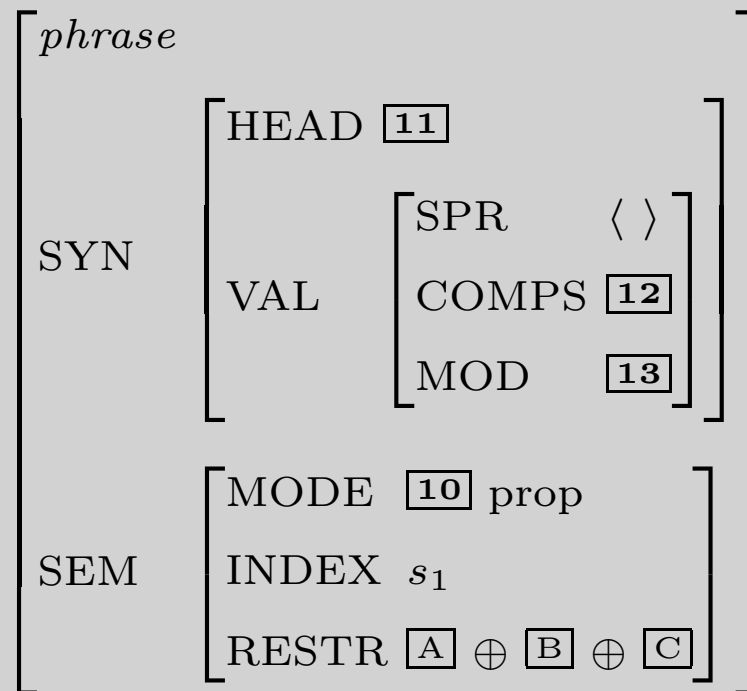
Key



Is this description fully specified?



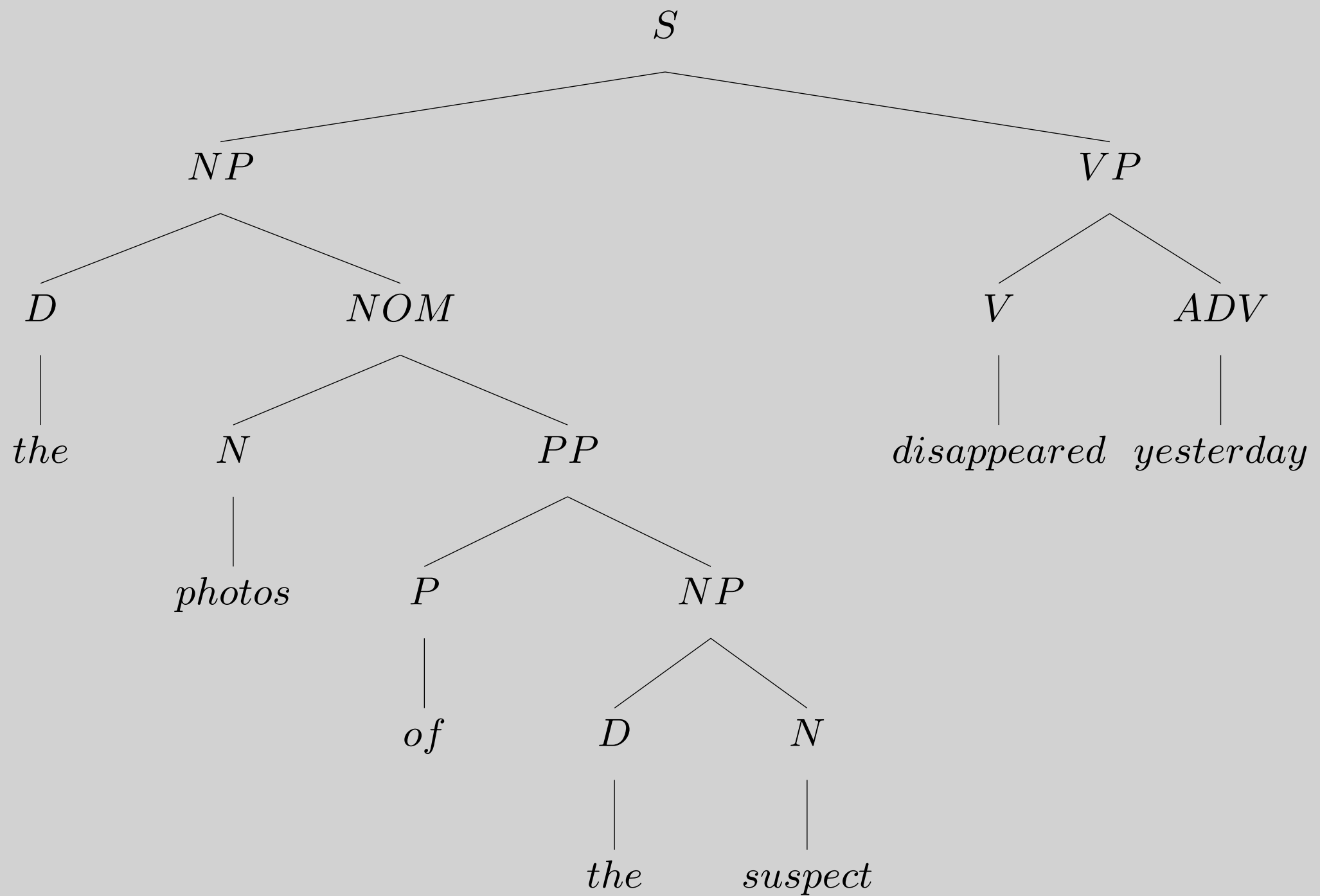
Does the top node satisfy the initial symbol?



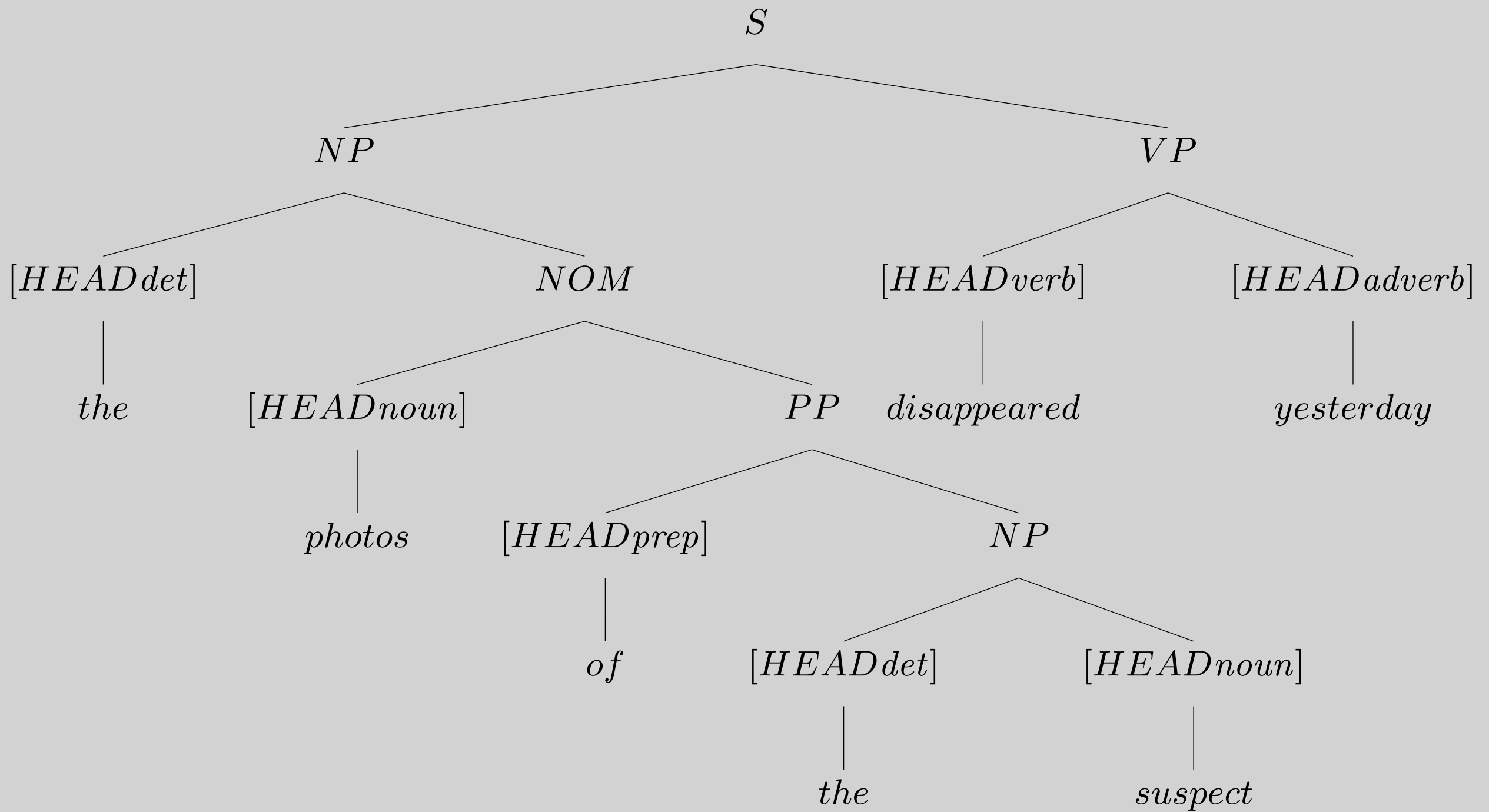
RESTR of the S node

$$\left\langle \begin{bmatrix} \text{RELN} & a \\ \text{BV} & k \end{bmatrix}, \begin{bmatrix} \text{RELN} & \text{cat} \\ \text{INST} & k \end{bmatrix}, \begin{bmatrix} \text{RELN} & \text{sleep} \\ \text{SIT} & s_1 \\ \text{SLEEPER} & k \end{bmatrix}, \dots \right\rangle$$

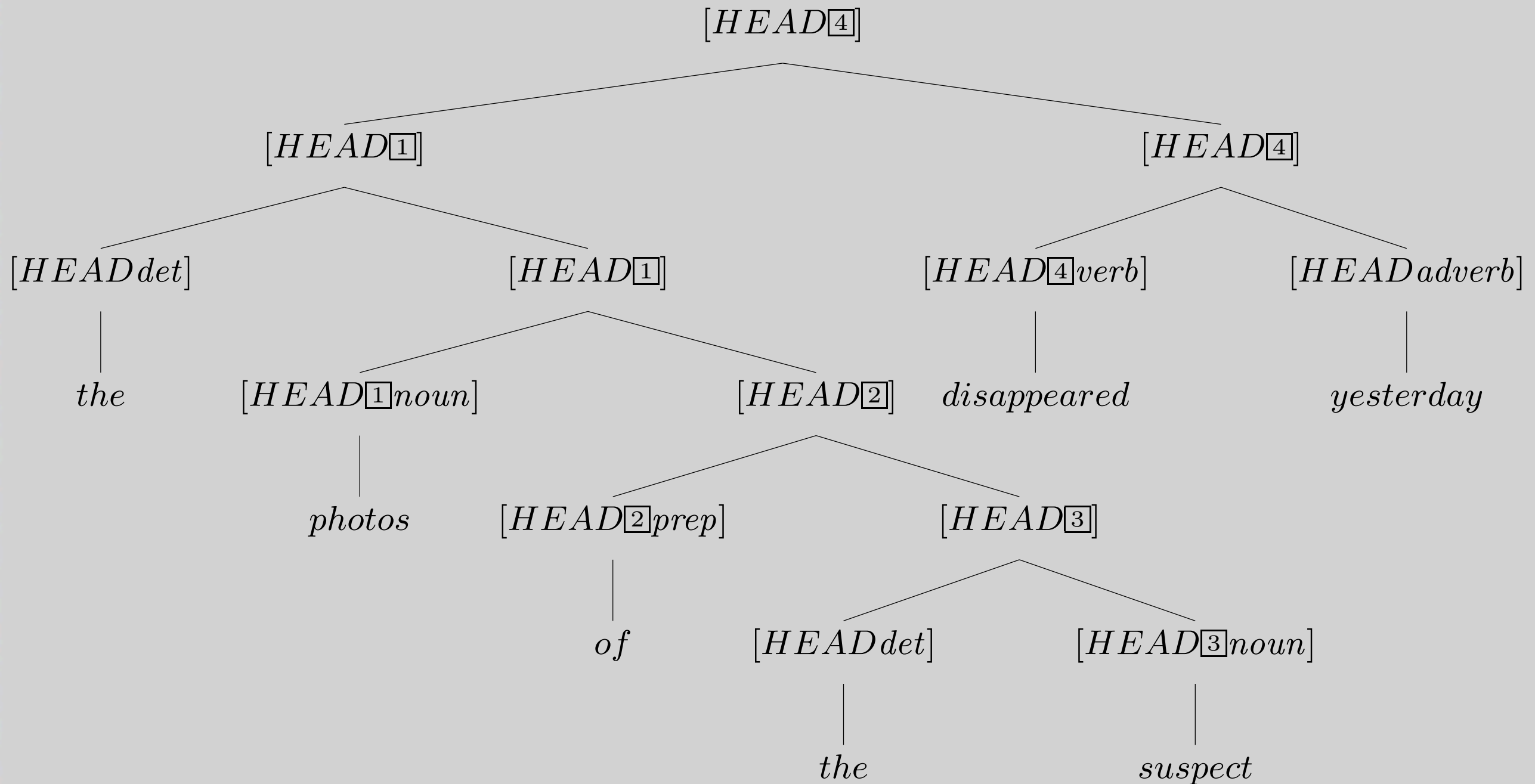
Another Example



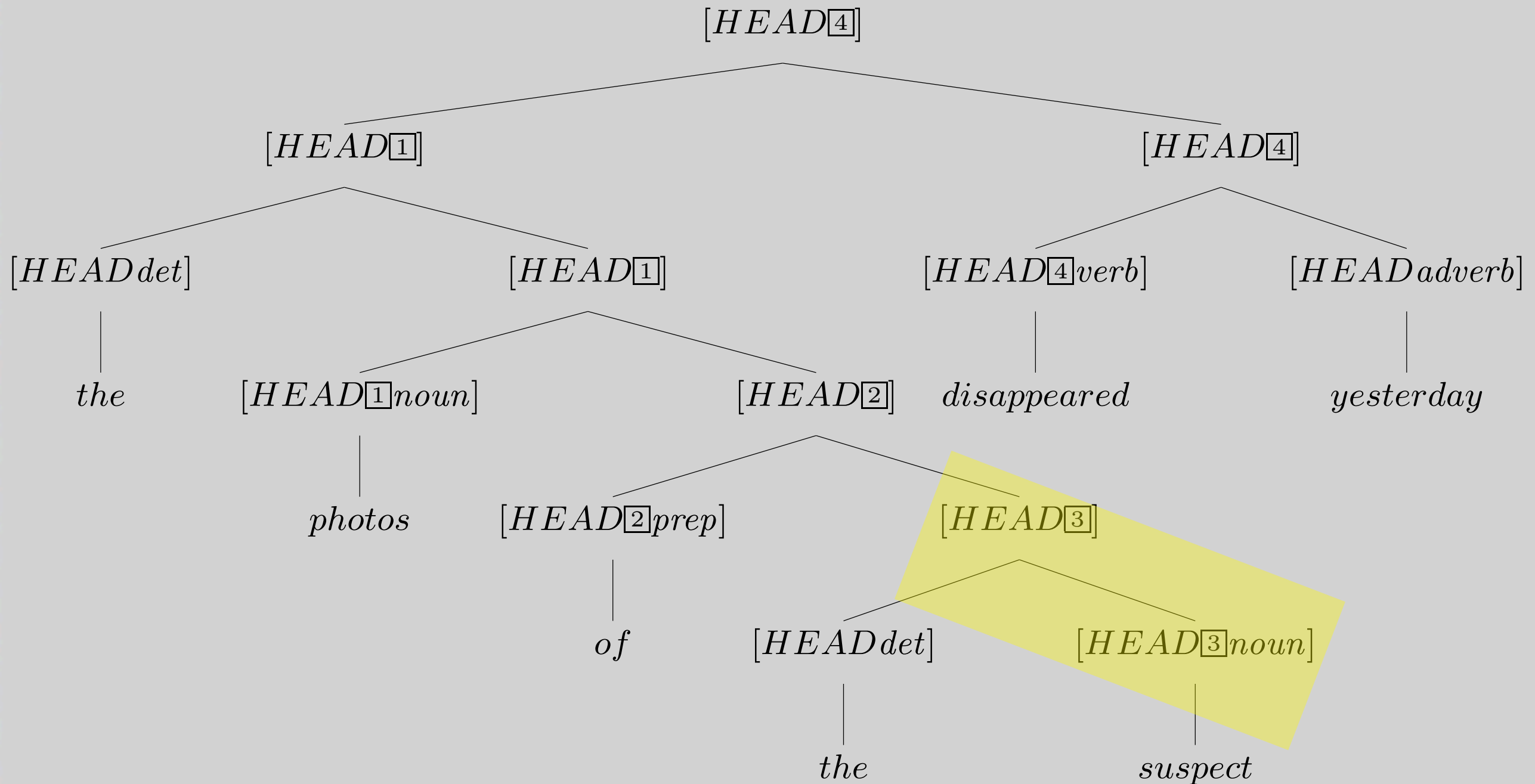
Head Features from Lexical Entries



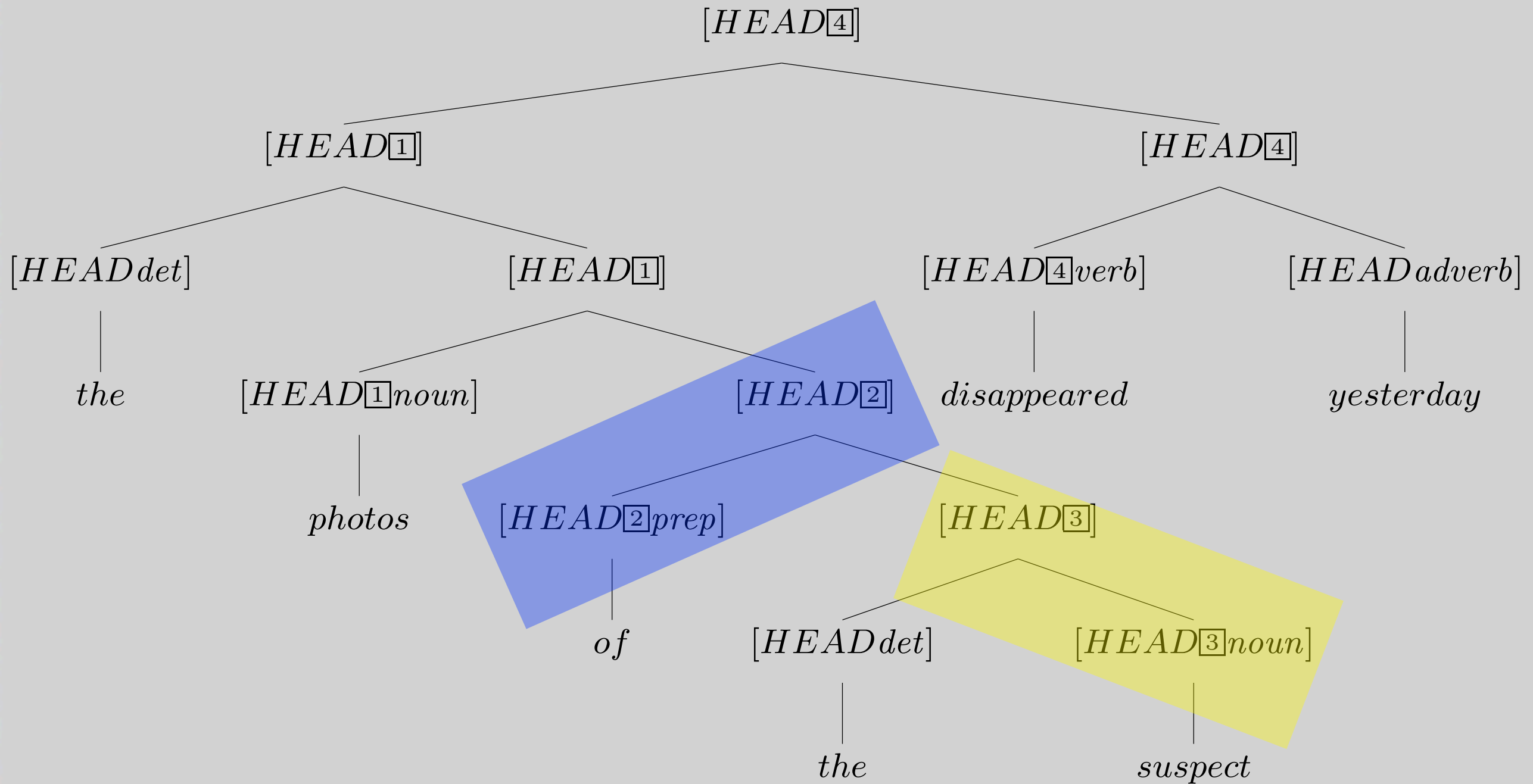
Head Features from Lexical Entries, plus HFP



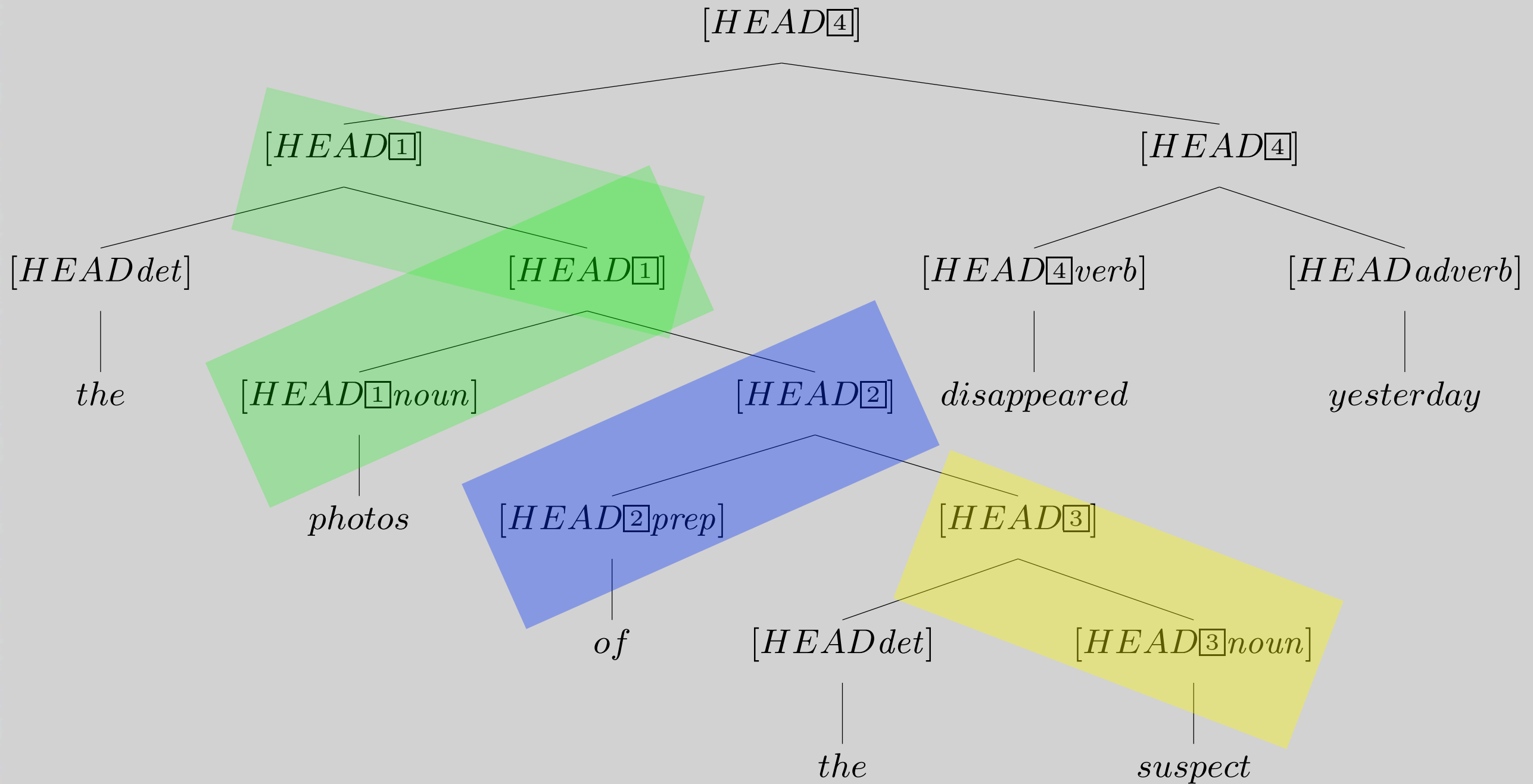
Head Features from Lexical Entries, plus HFP



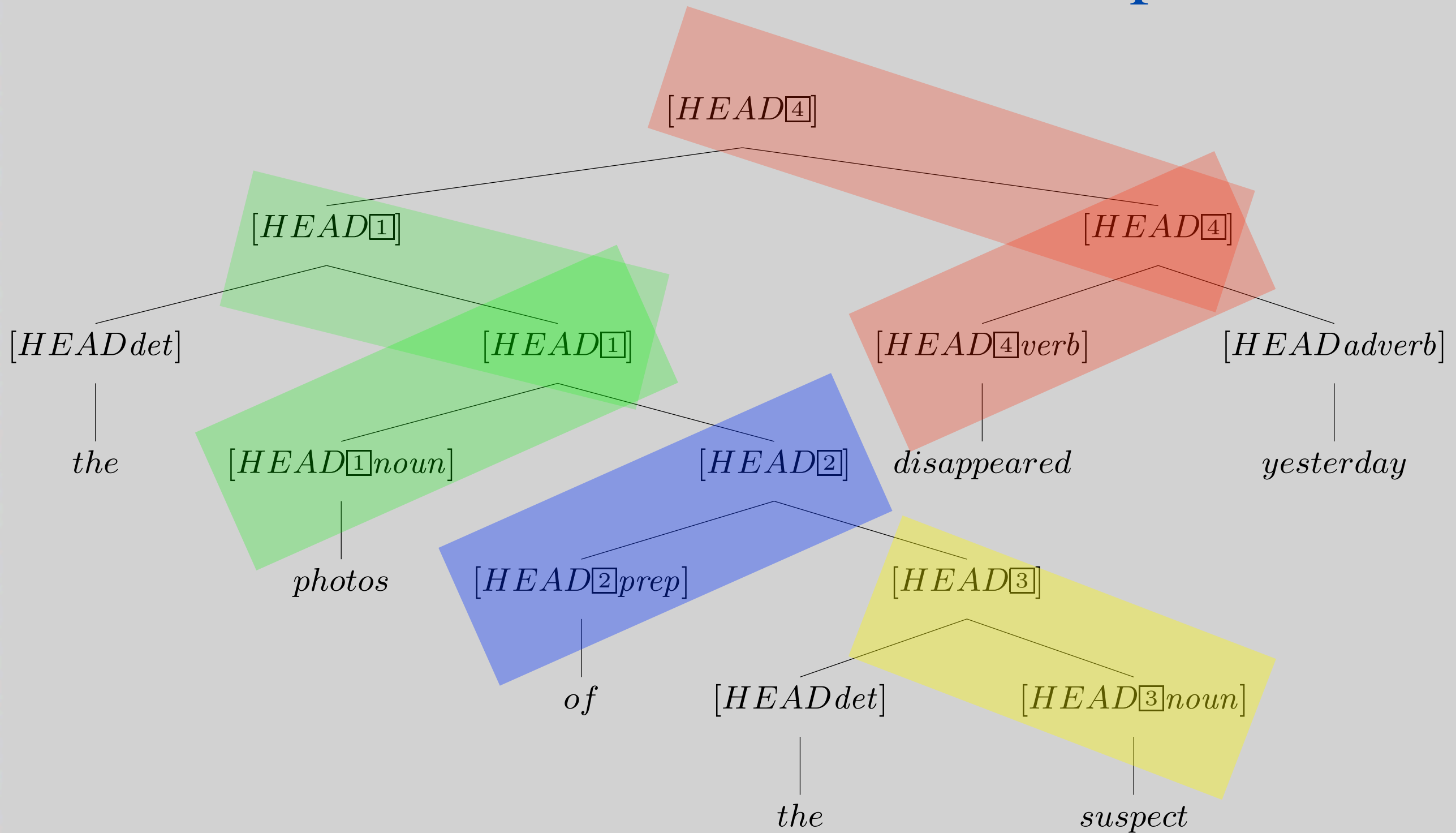
Head Features from Lexical Entries, plus HFP



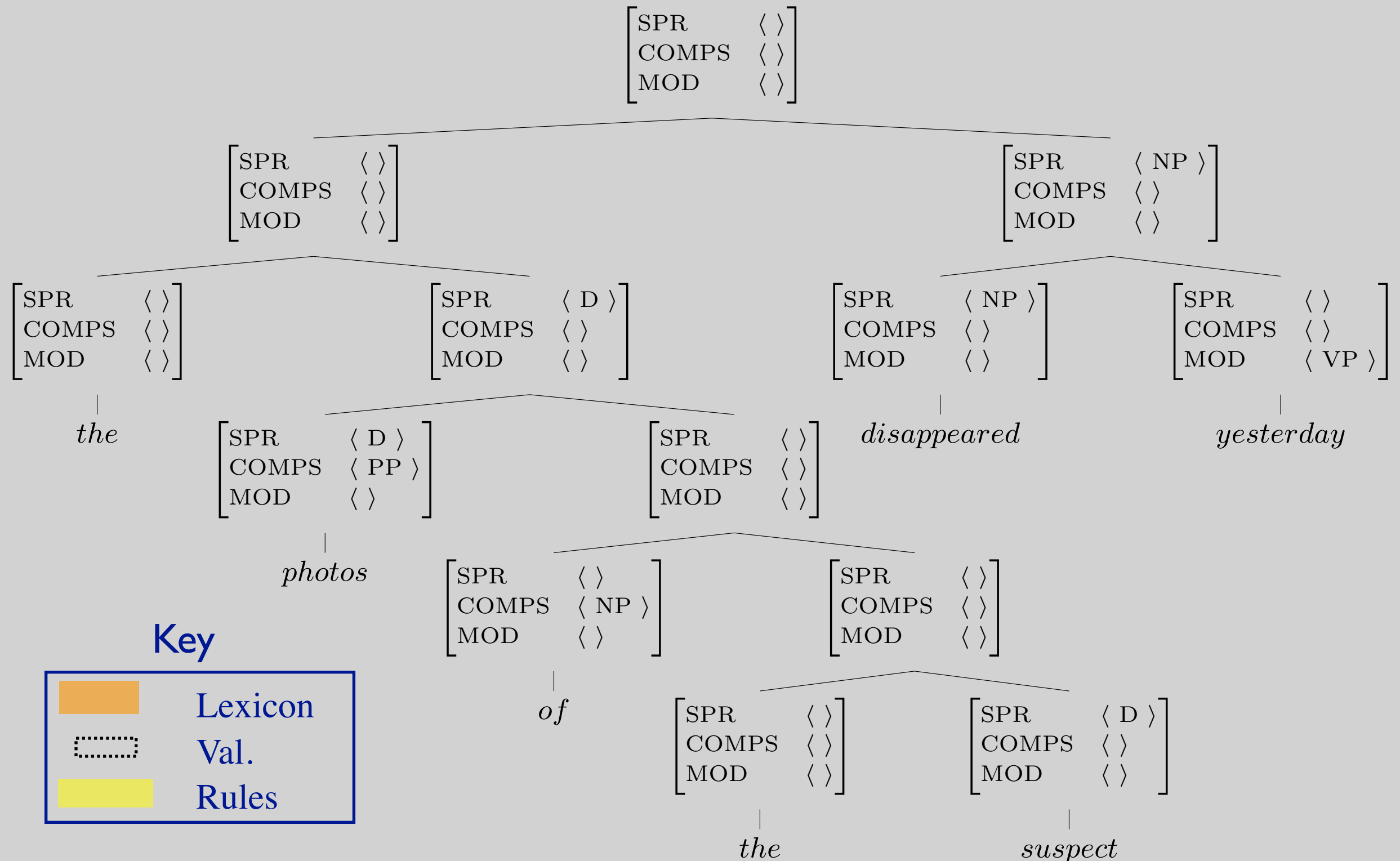
Head Features from Lexical Entries, plus HFP



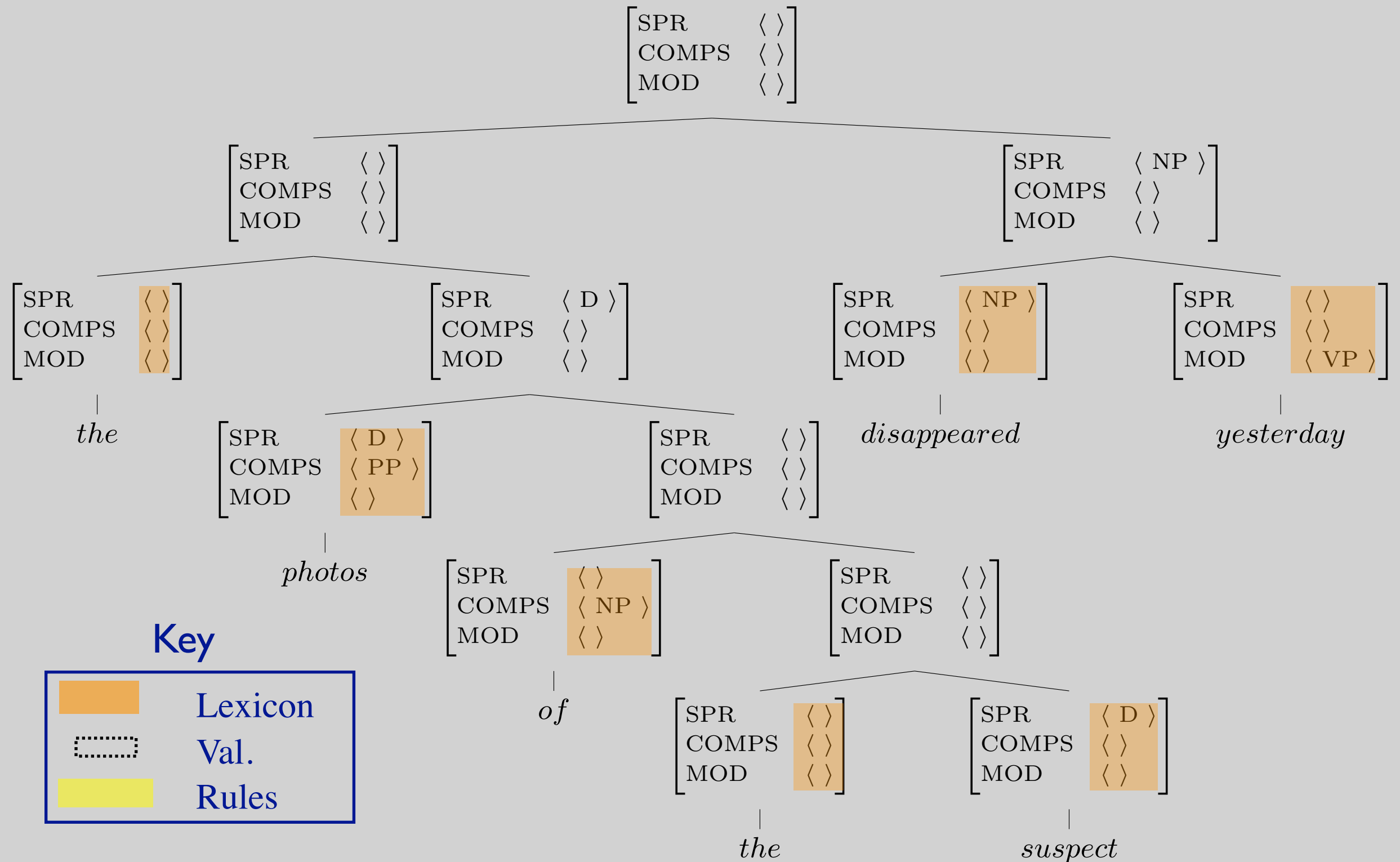
Head Features from Lexical Entries, plus HFP



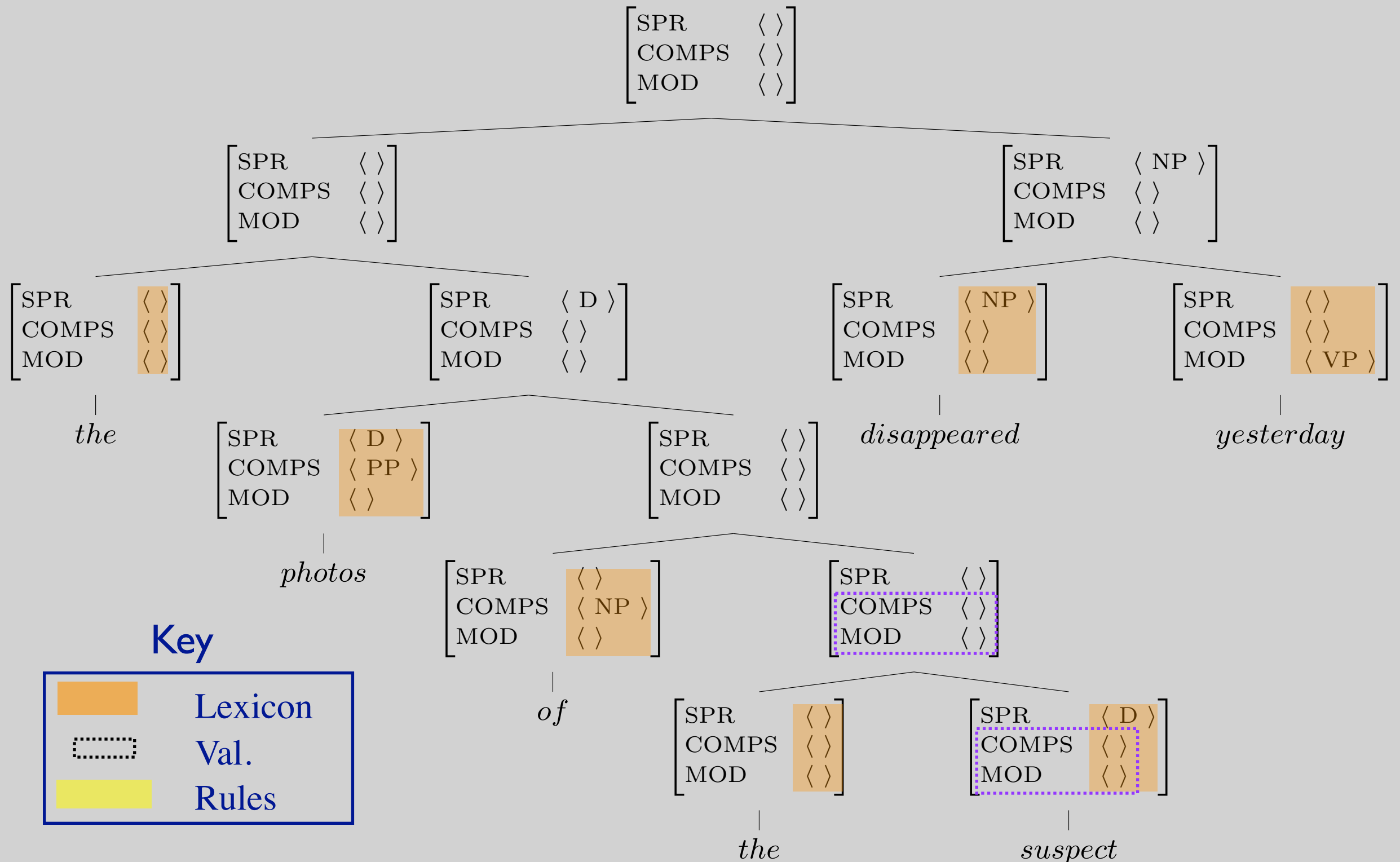
Valence Features: Lexicon, Rules, and the Valence Principle



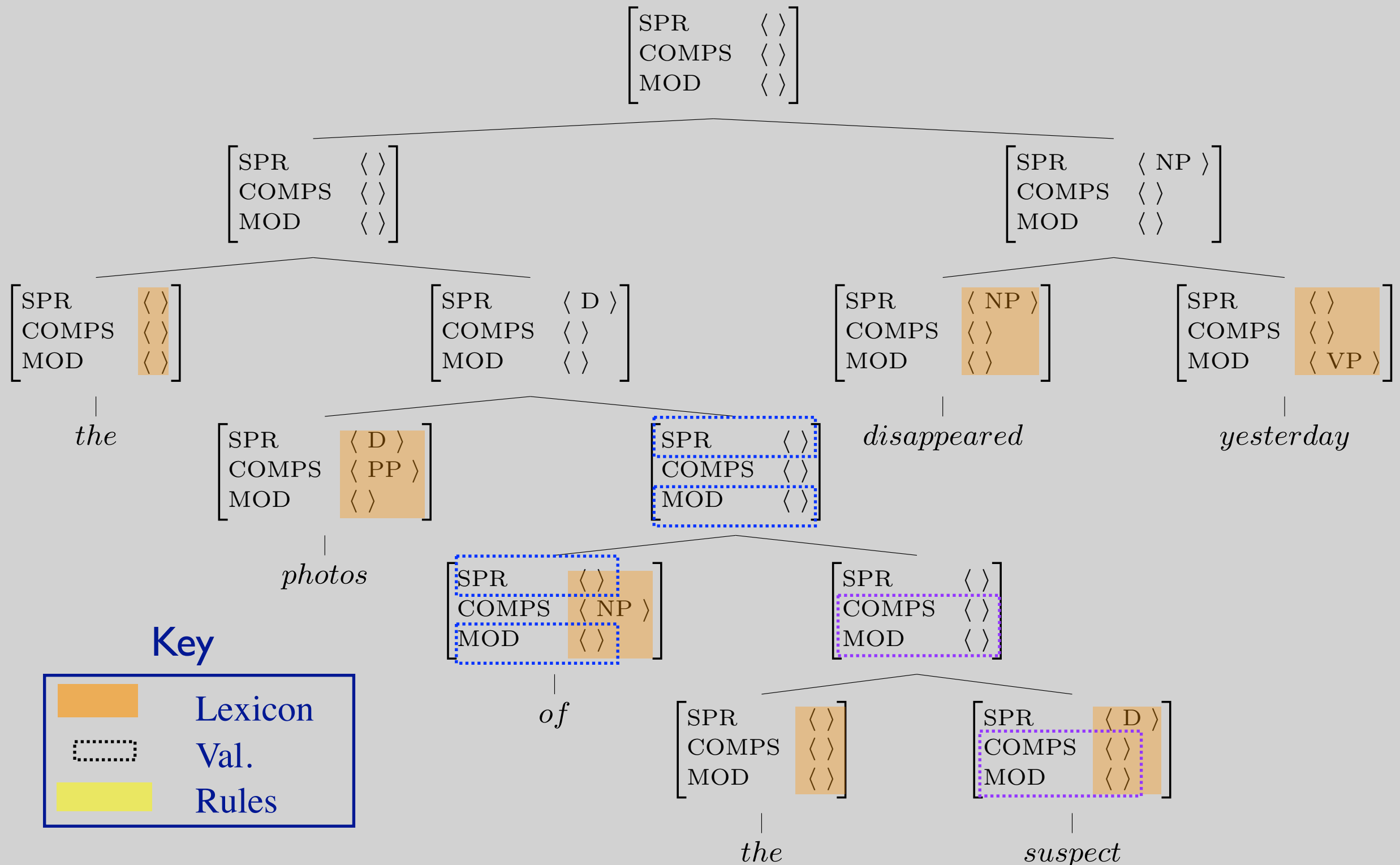
Valence Features: Lexicon, Rules, and the Valence Principle



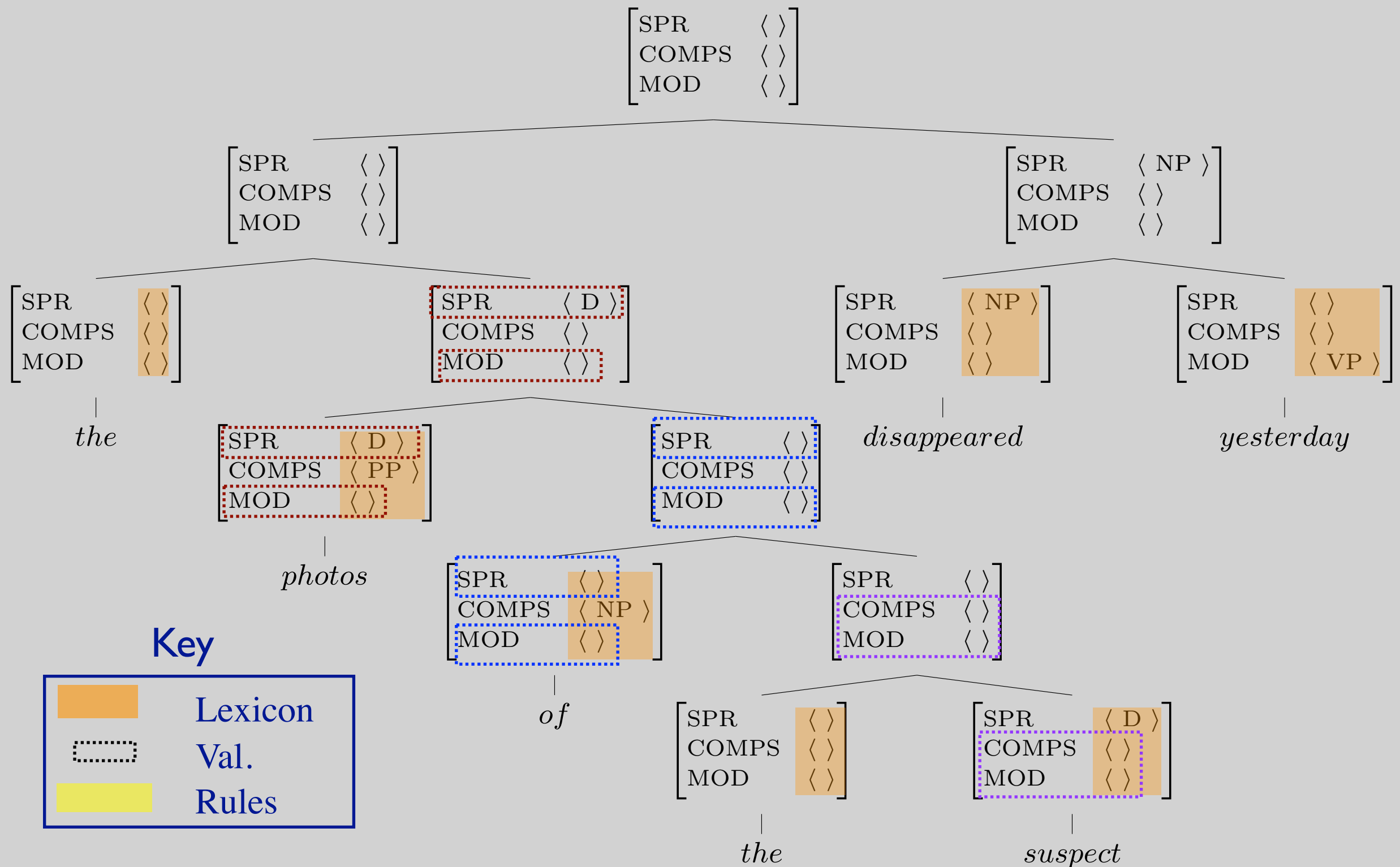
Valence Features: Lexicon, Rules, and the Valence Principle



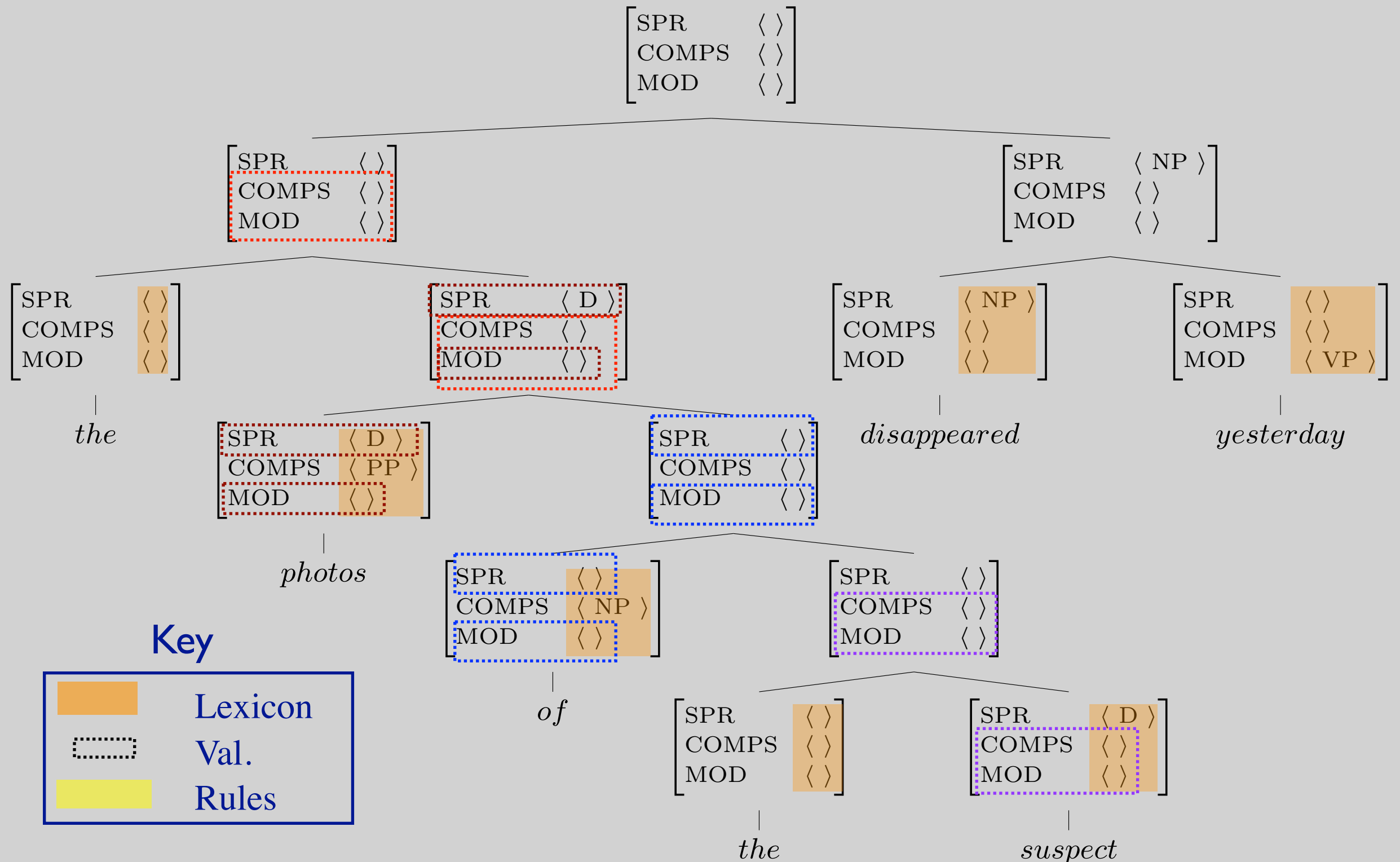
Valence Features: Lexicon, Rules, and the Valence Principle



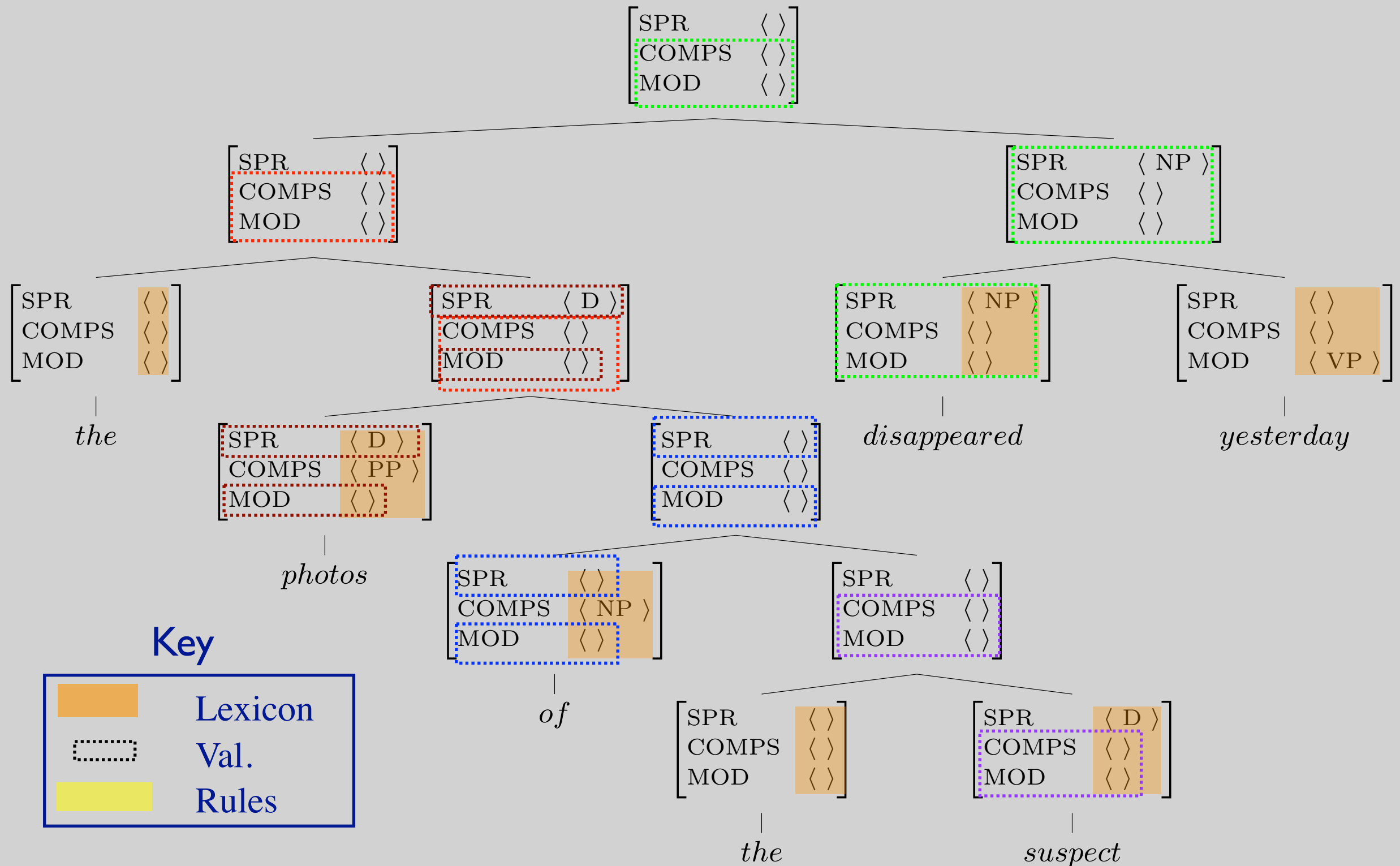
Valence Features: Lexicon, Rules, and the Valence Principle



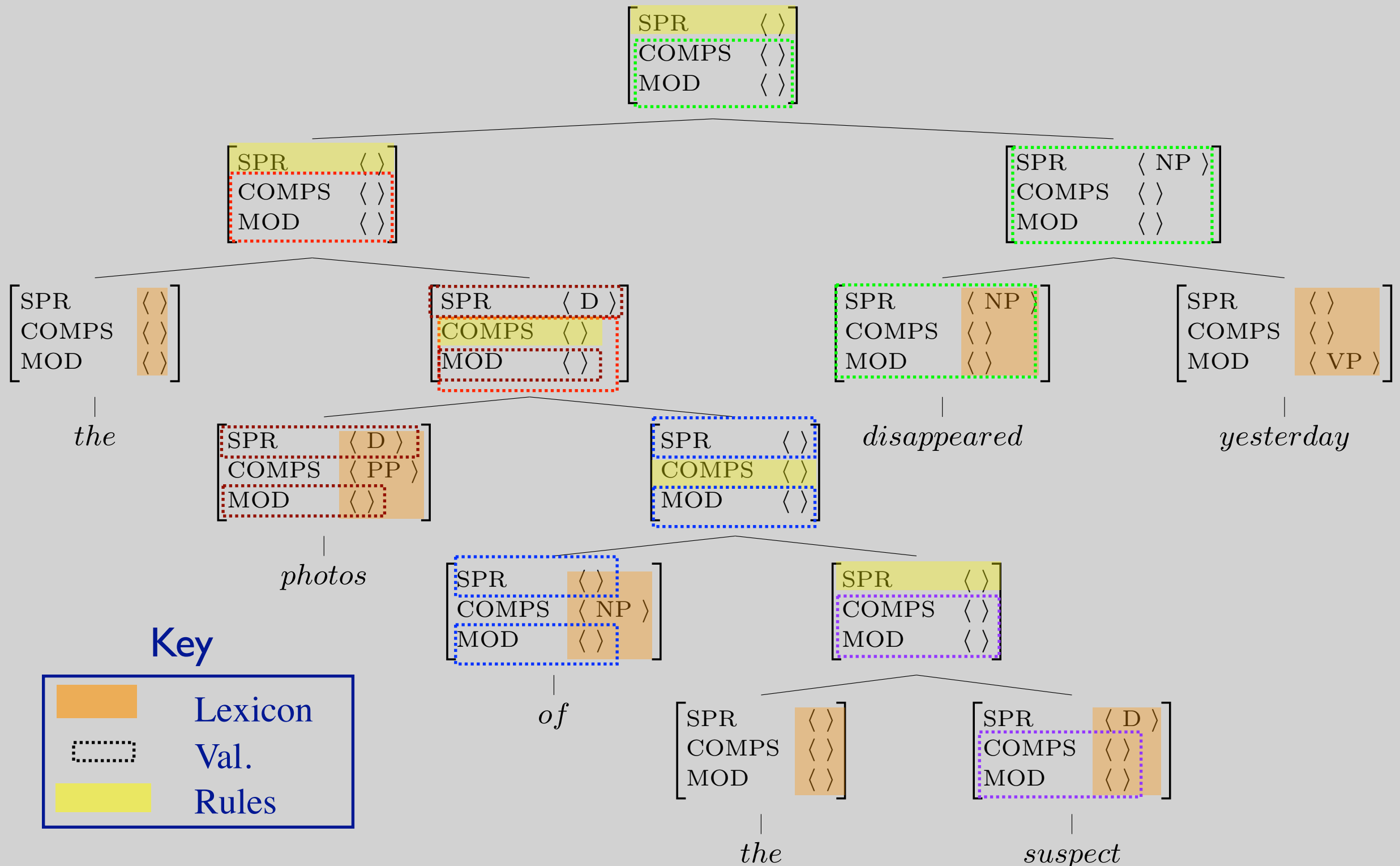
Valence Features: Lexicon, Rules, and the Valence Principle



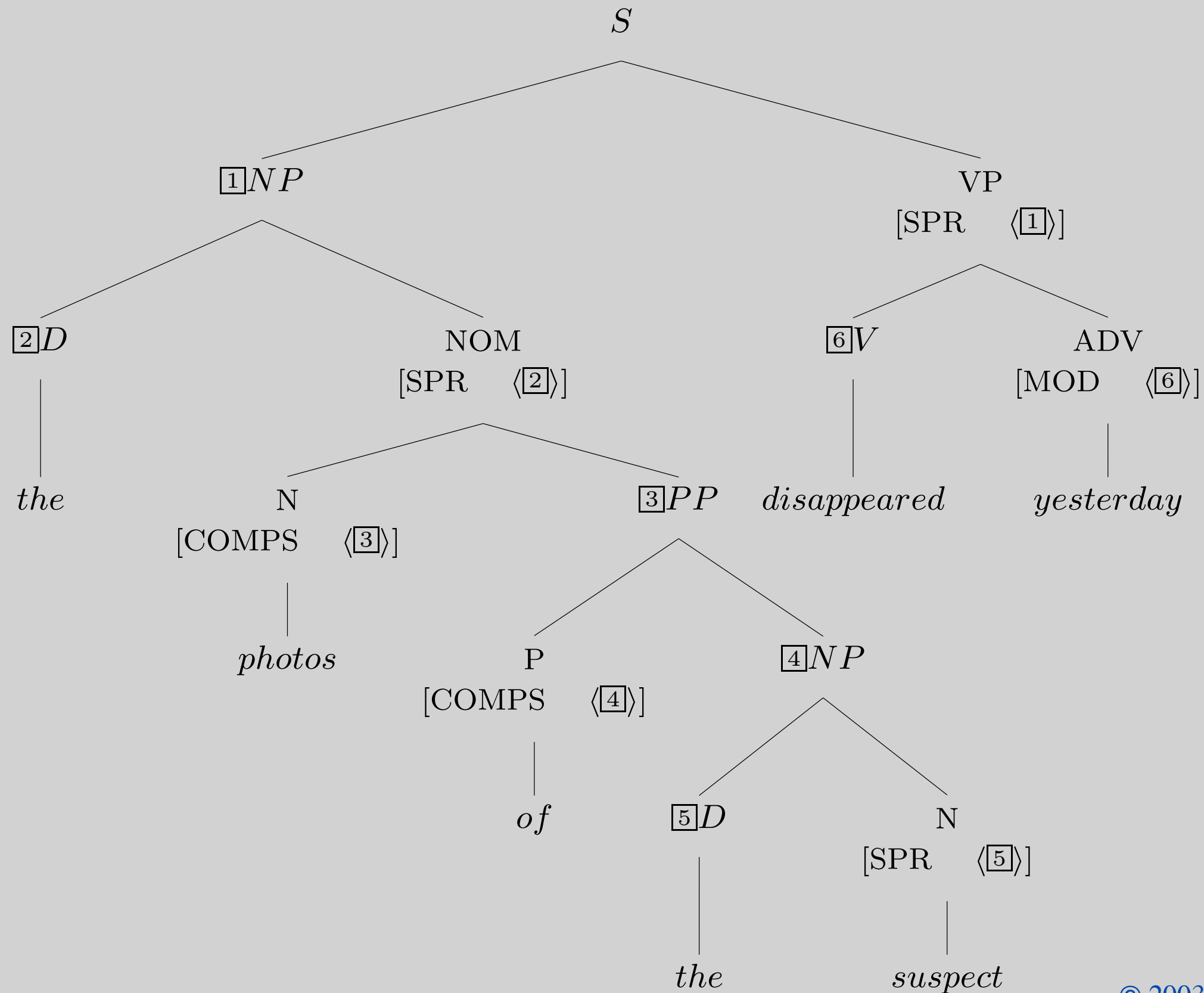
Valence Features: Lexicon, Rules, and the Valence Principle



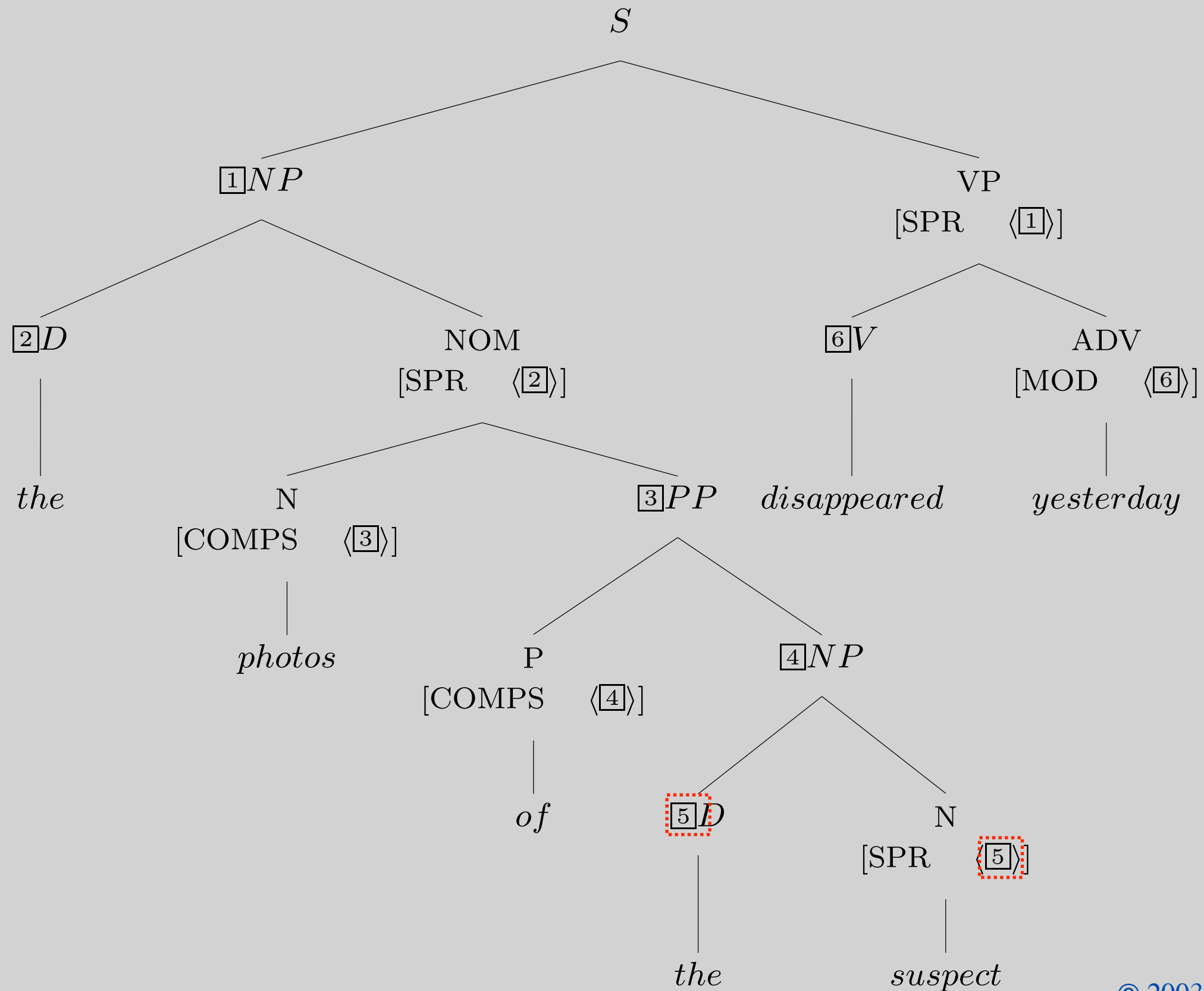
Valence Features: Lexicon, Rules, and the Valence Principle



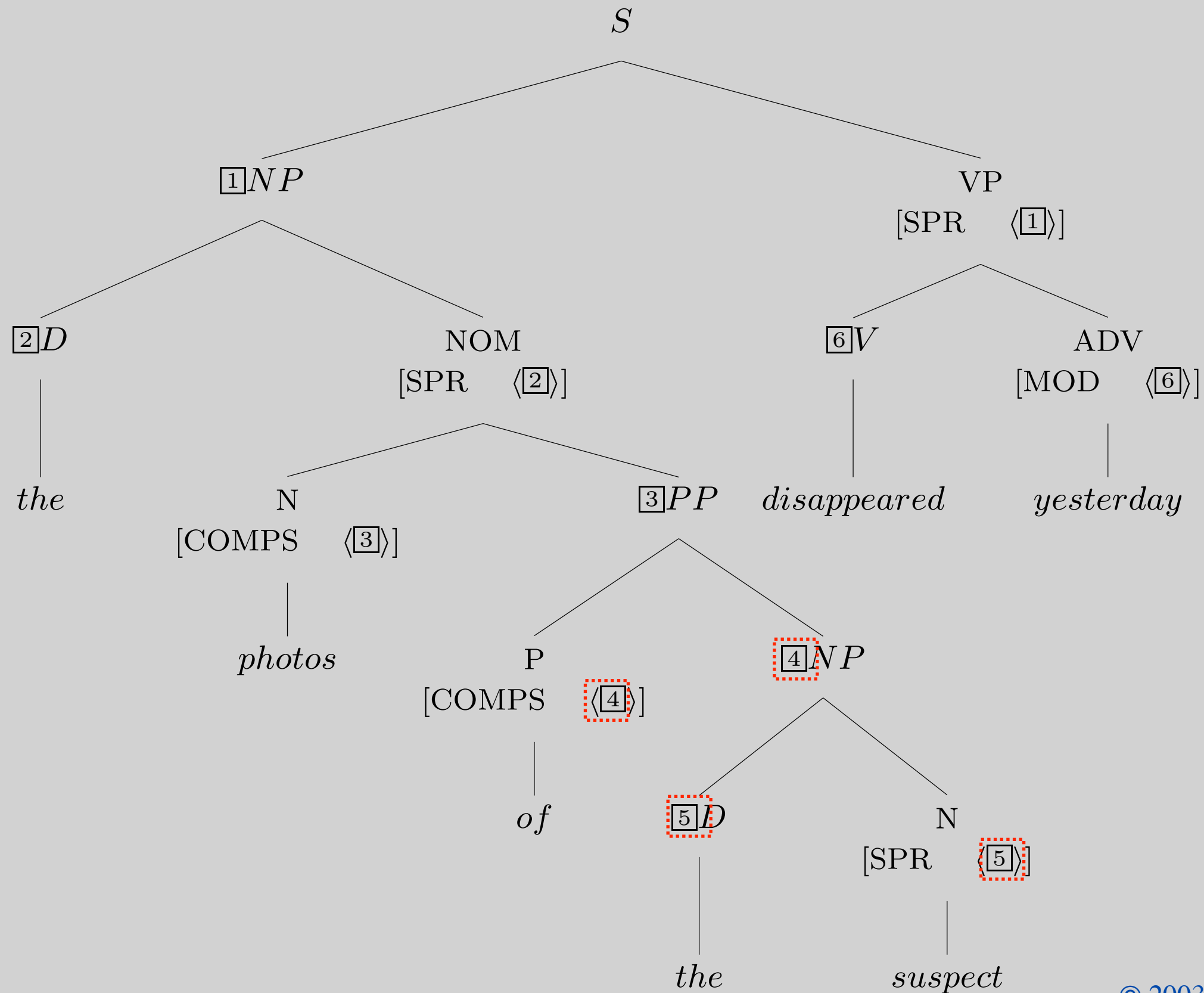
Required Identities: Grammar Rules



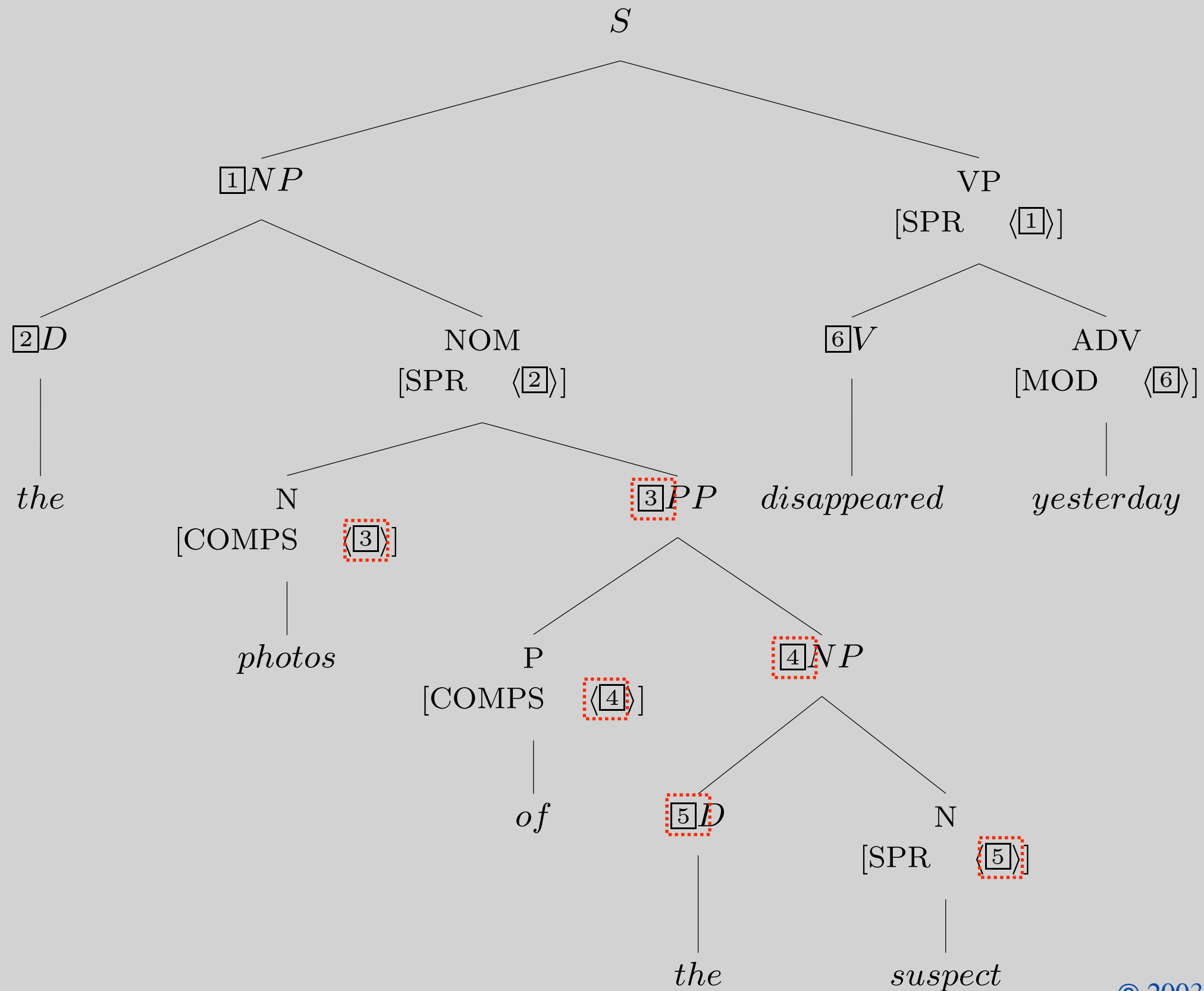
Required Identities: Grammar Rules



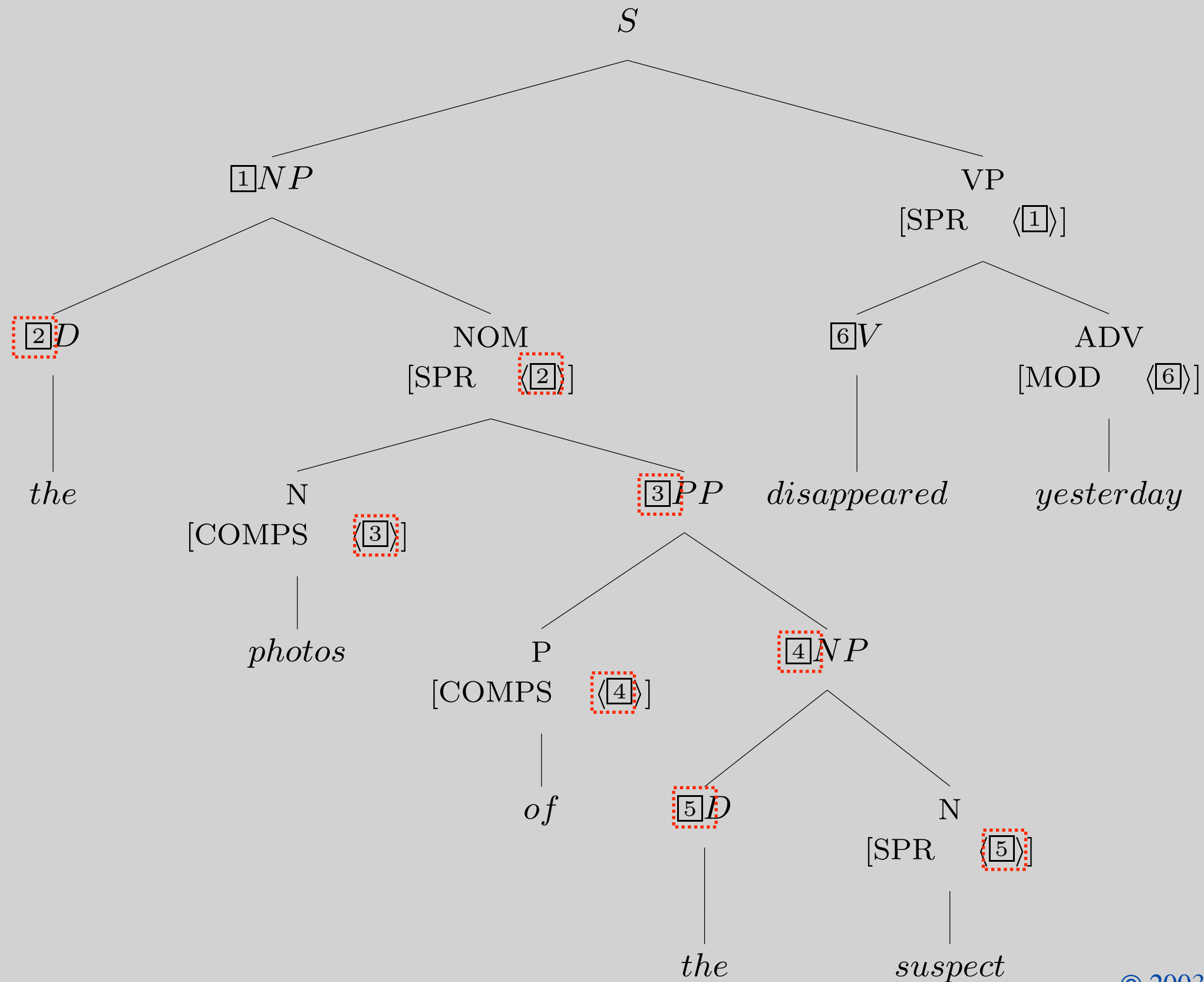
Required Identities: Grammar Rules



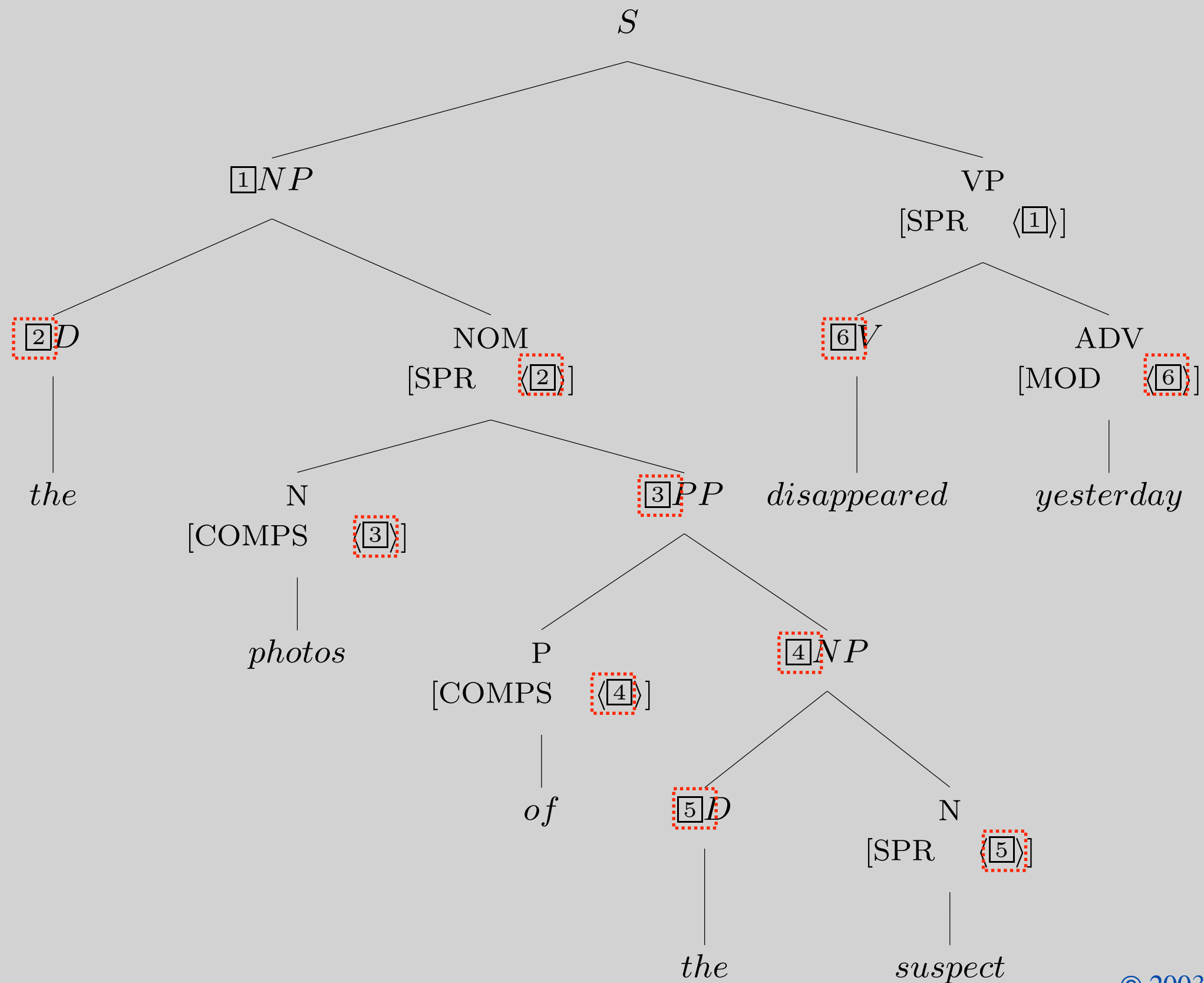
Required Identities: Grammar Rules



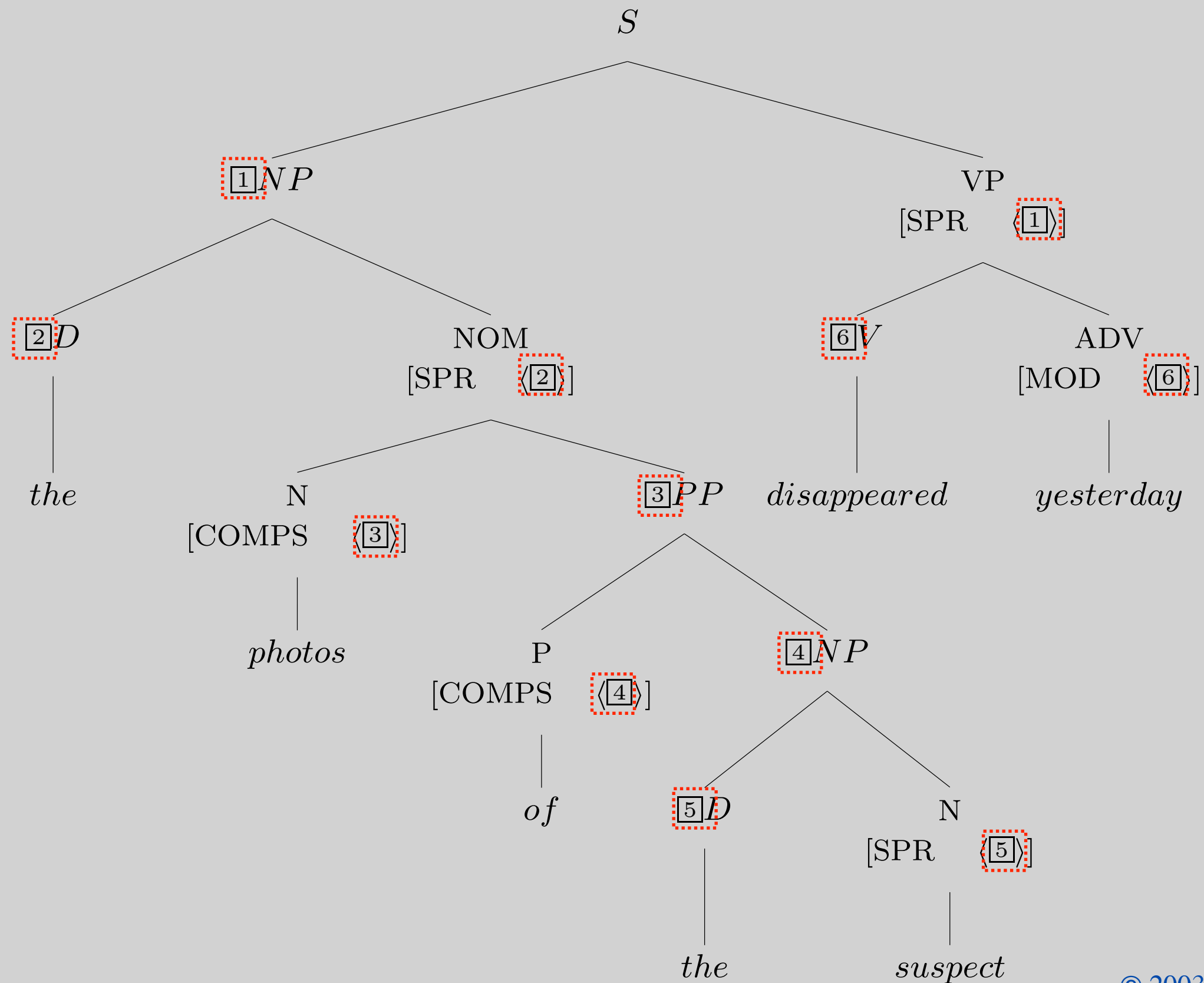
Required Identities: Grammar Rules



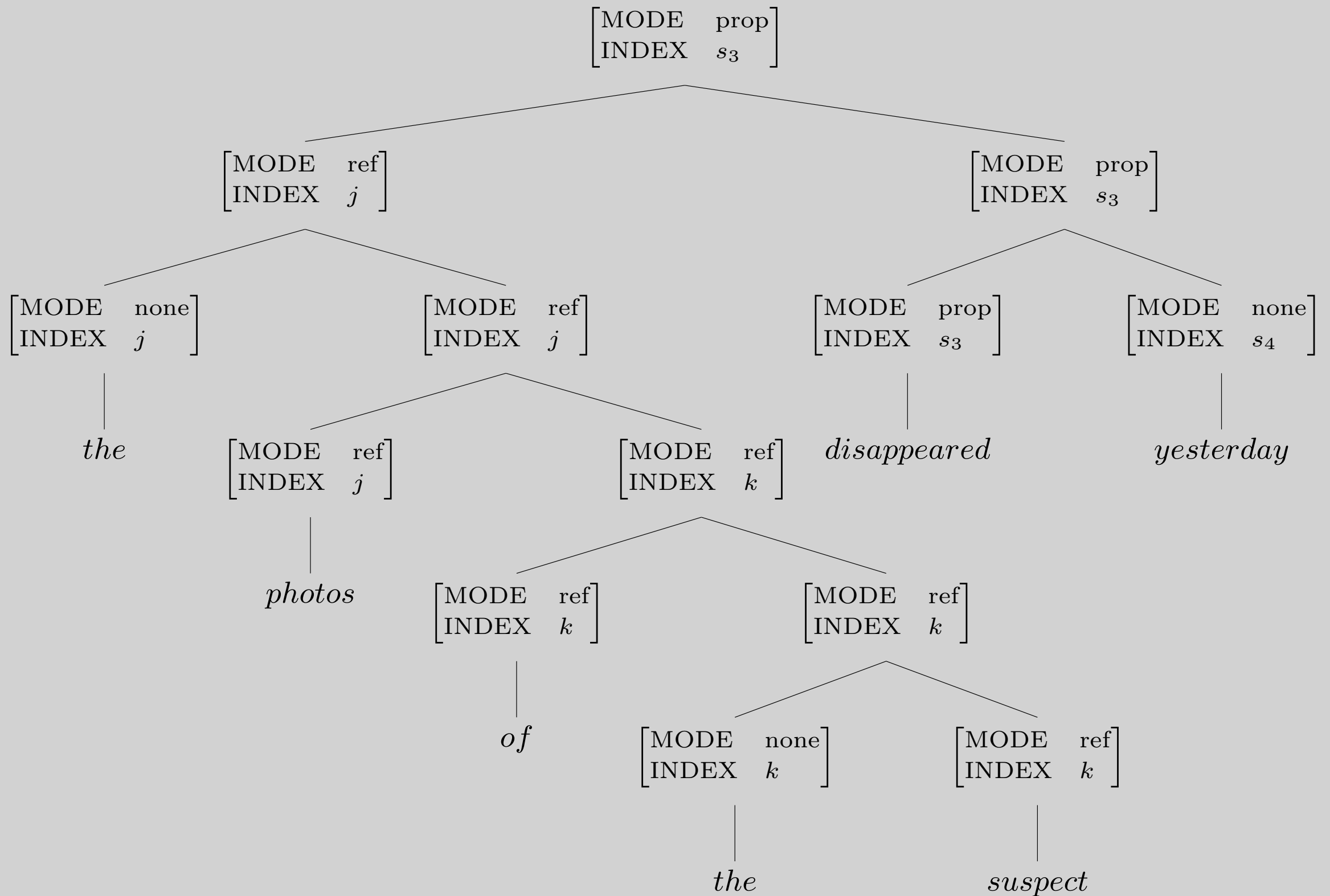
Required Identities: Grammar Rules



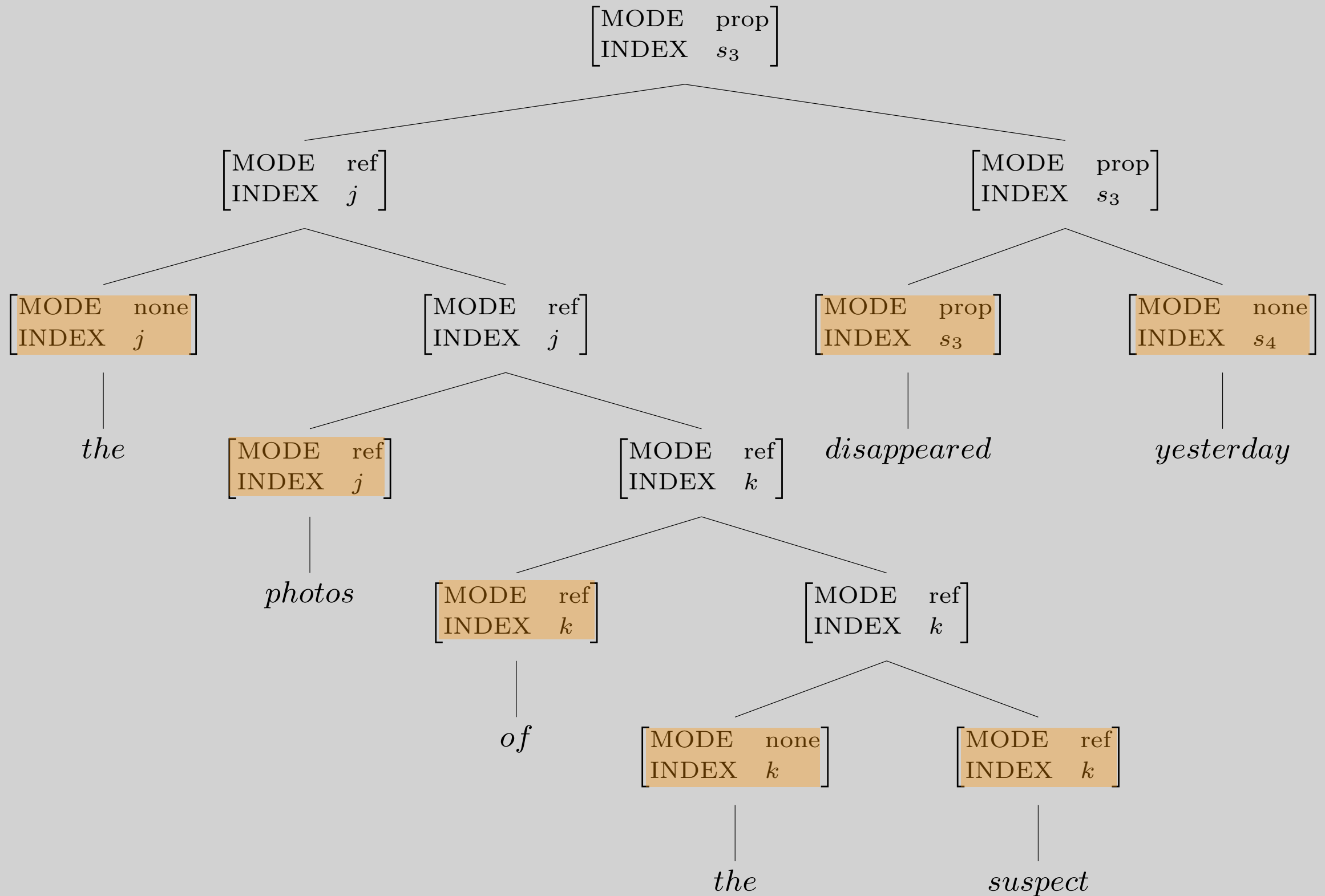
Required Identities: Grammar Rules



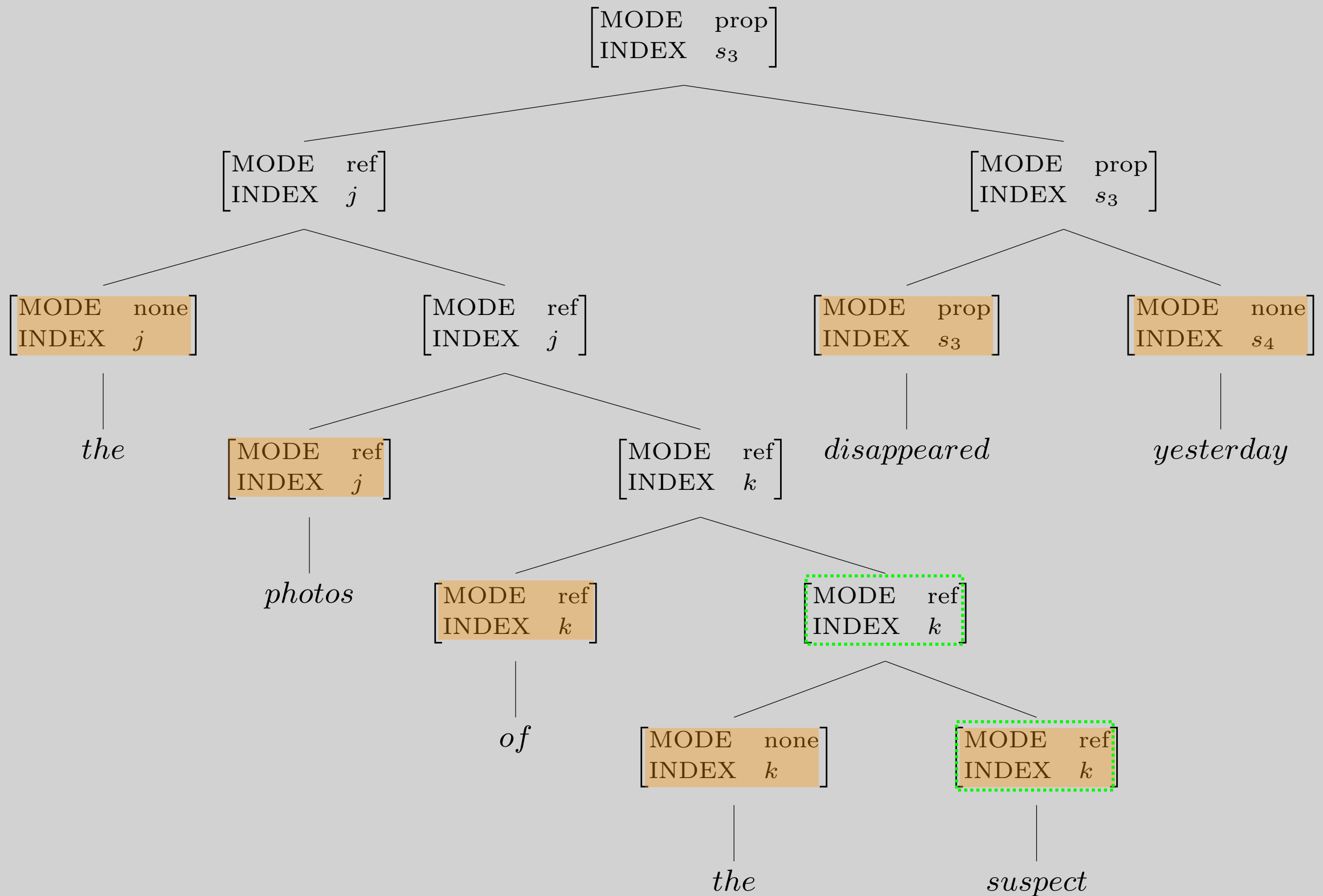
Two Semantic Features: the Lexicon & SIP



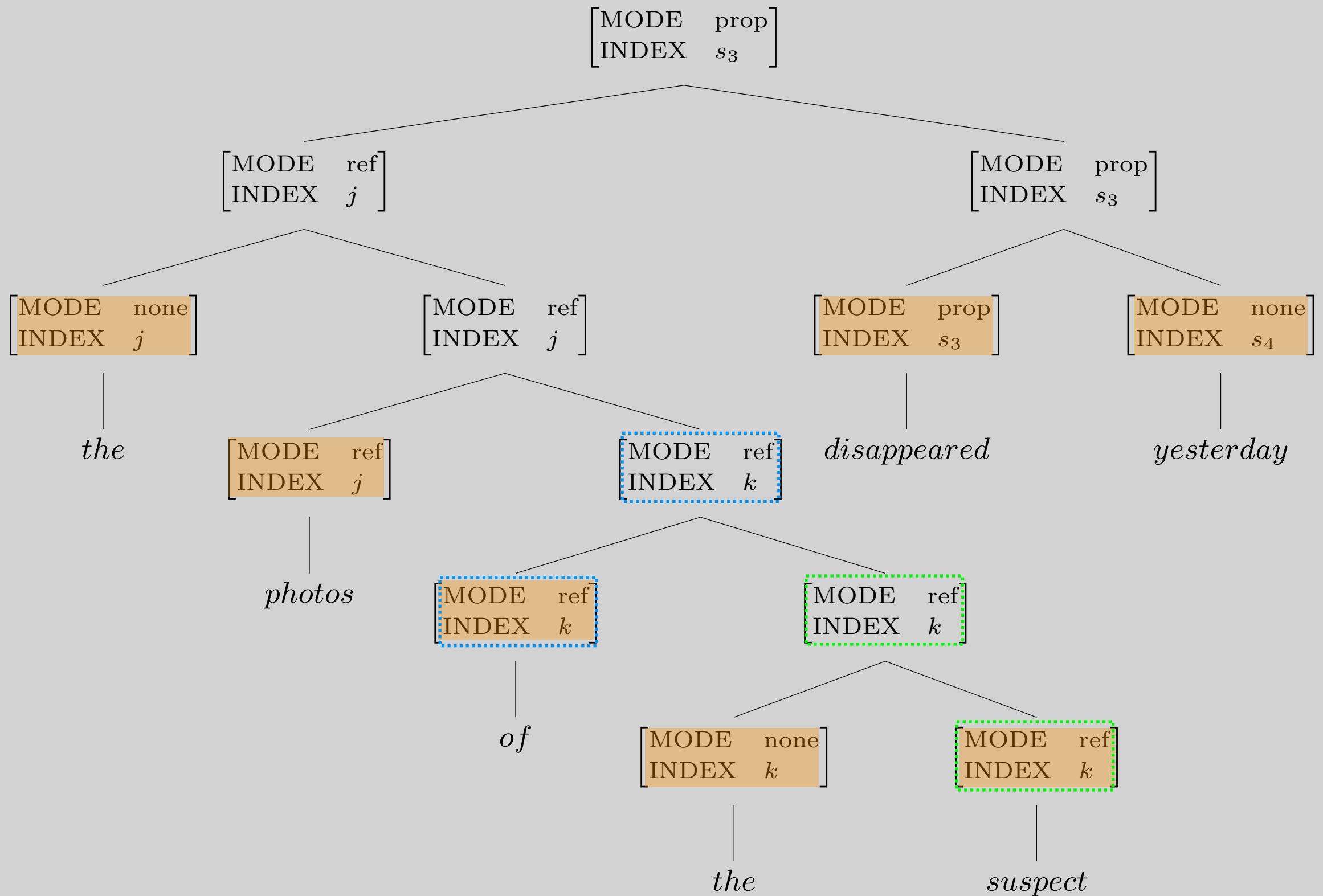
Two Semantic Features: the Lexicon & SIP



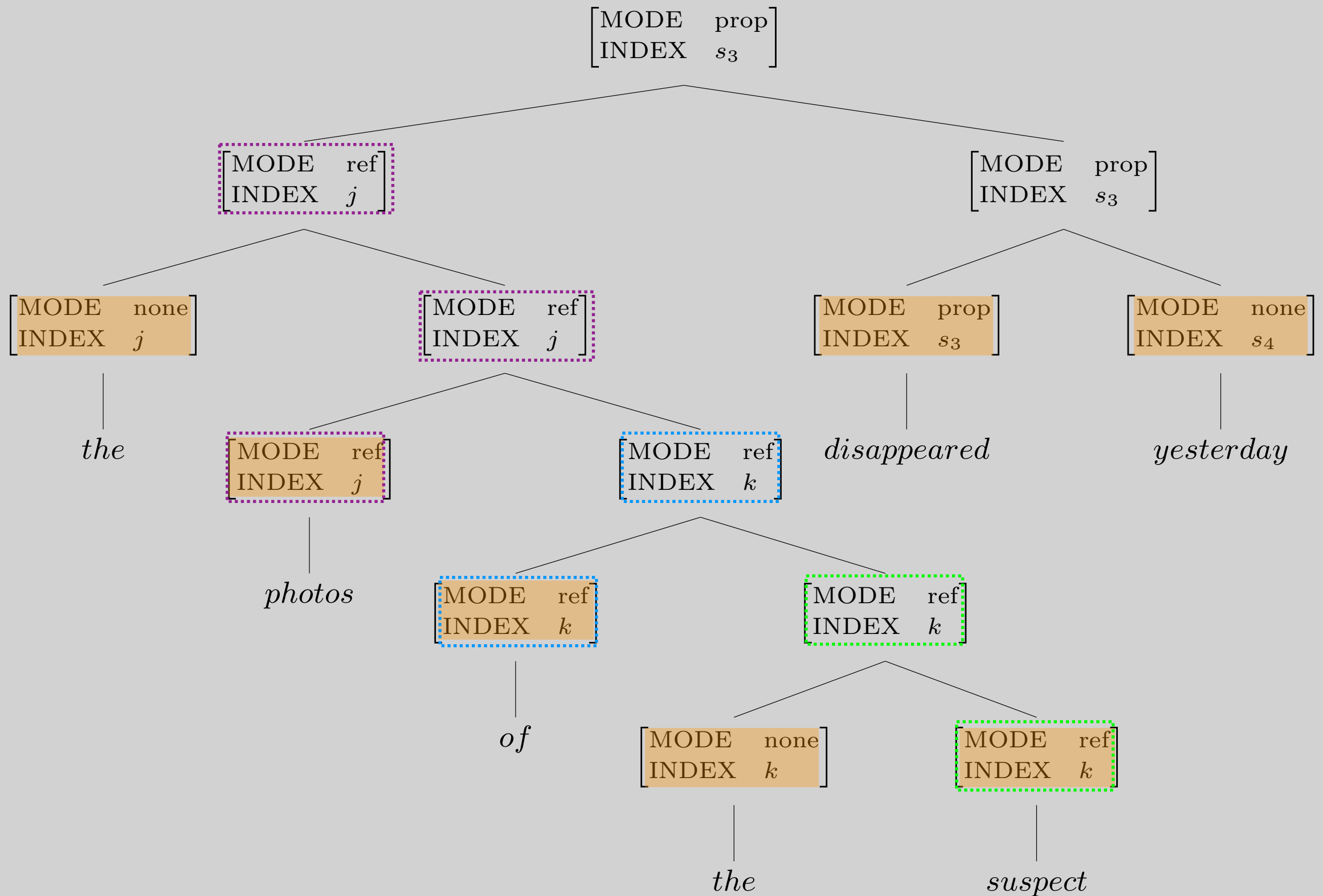
Two Semantic Features: the Lexicon & SIP



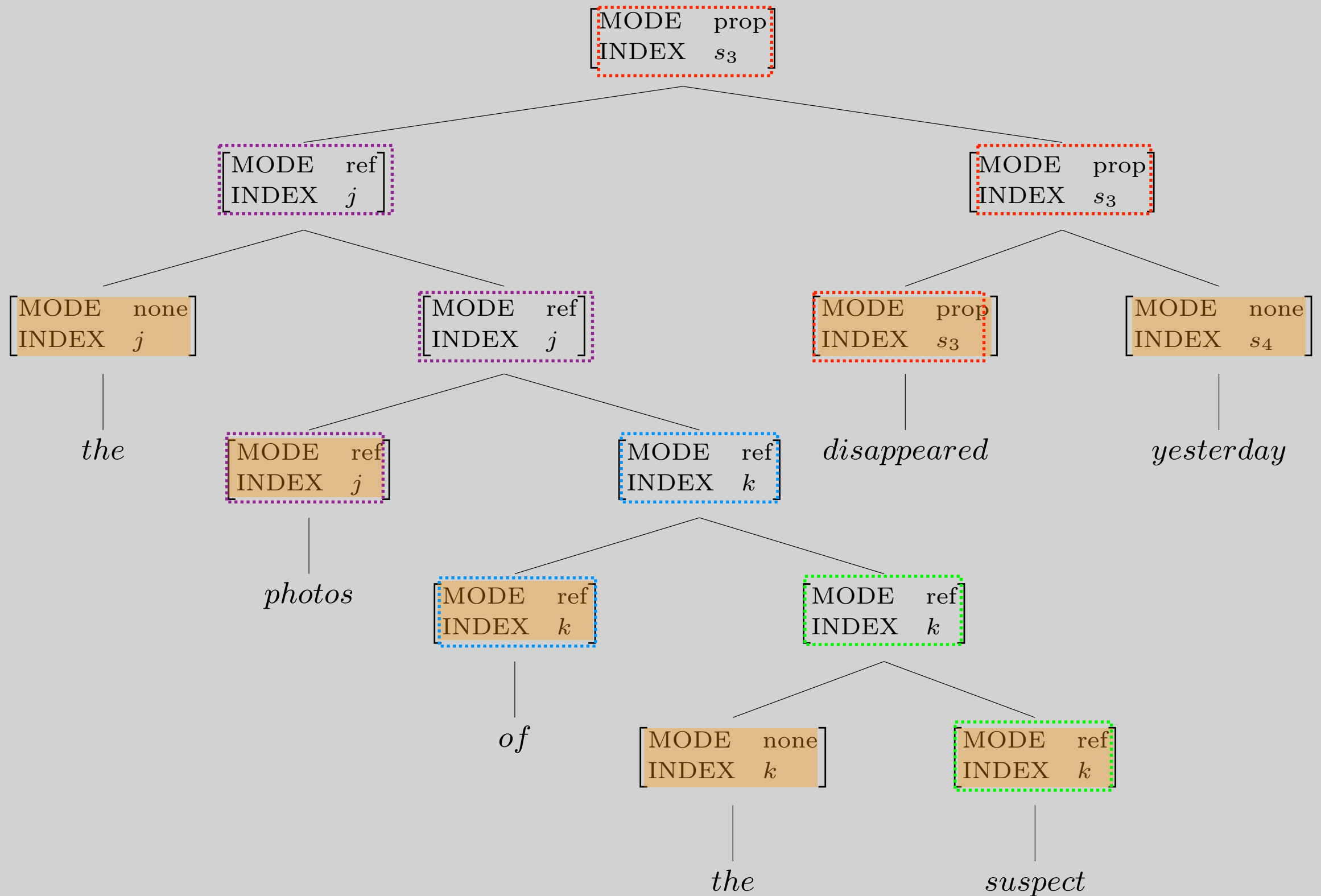
Two Semantic Features: the Lexicon & SIP



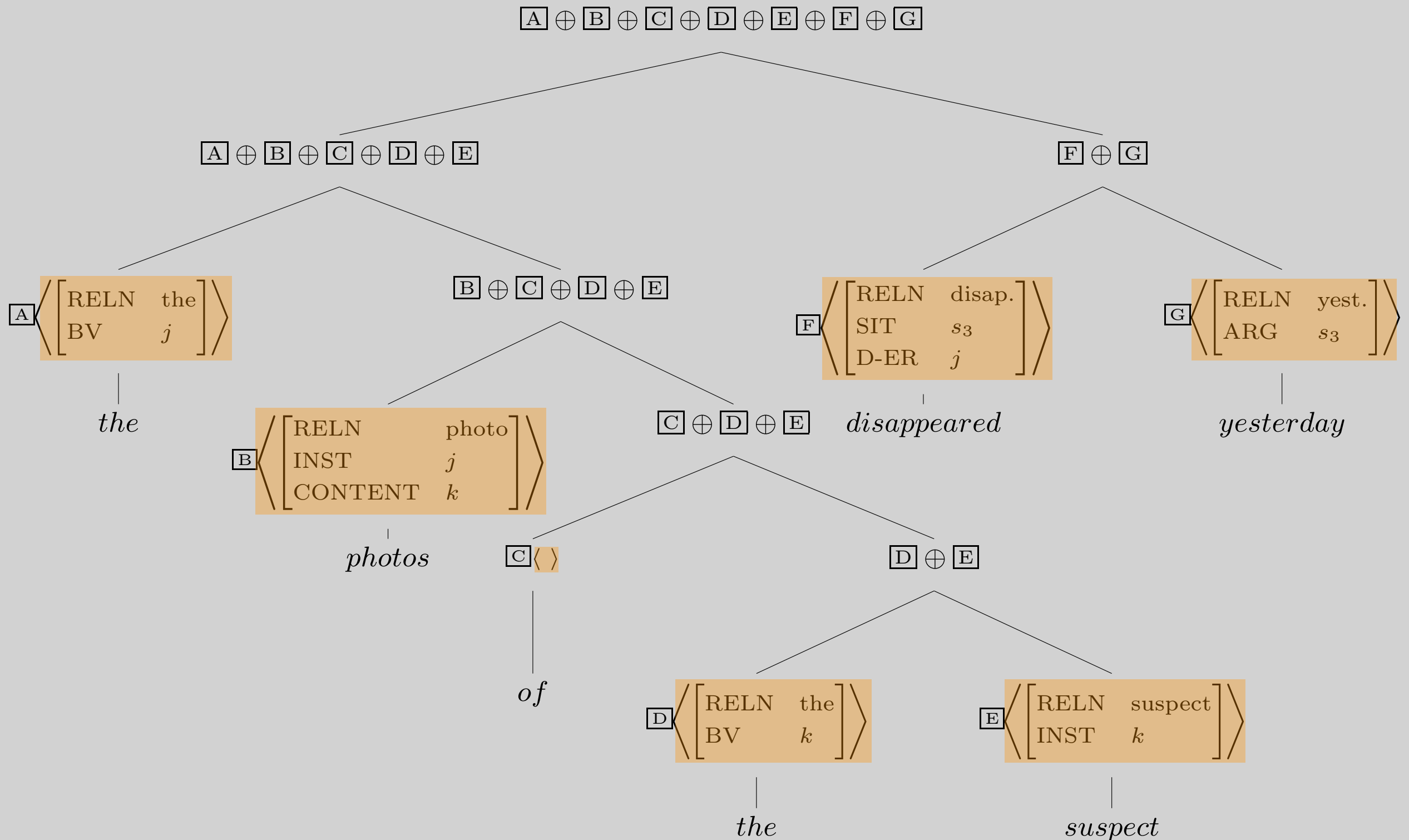
Two Semantic Features: the Lexicon & SIP



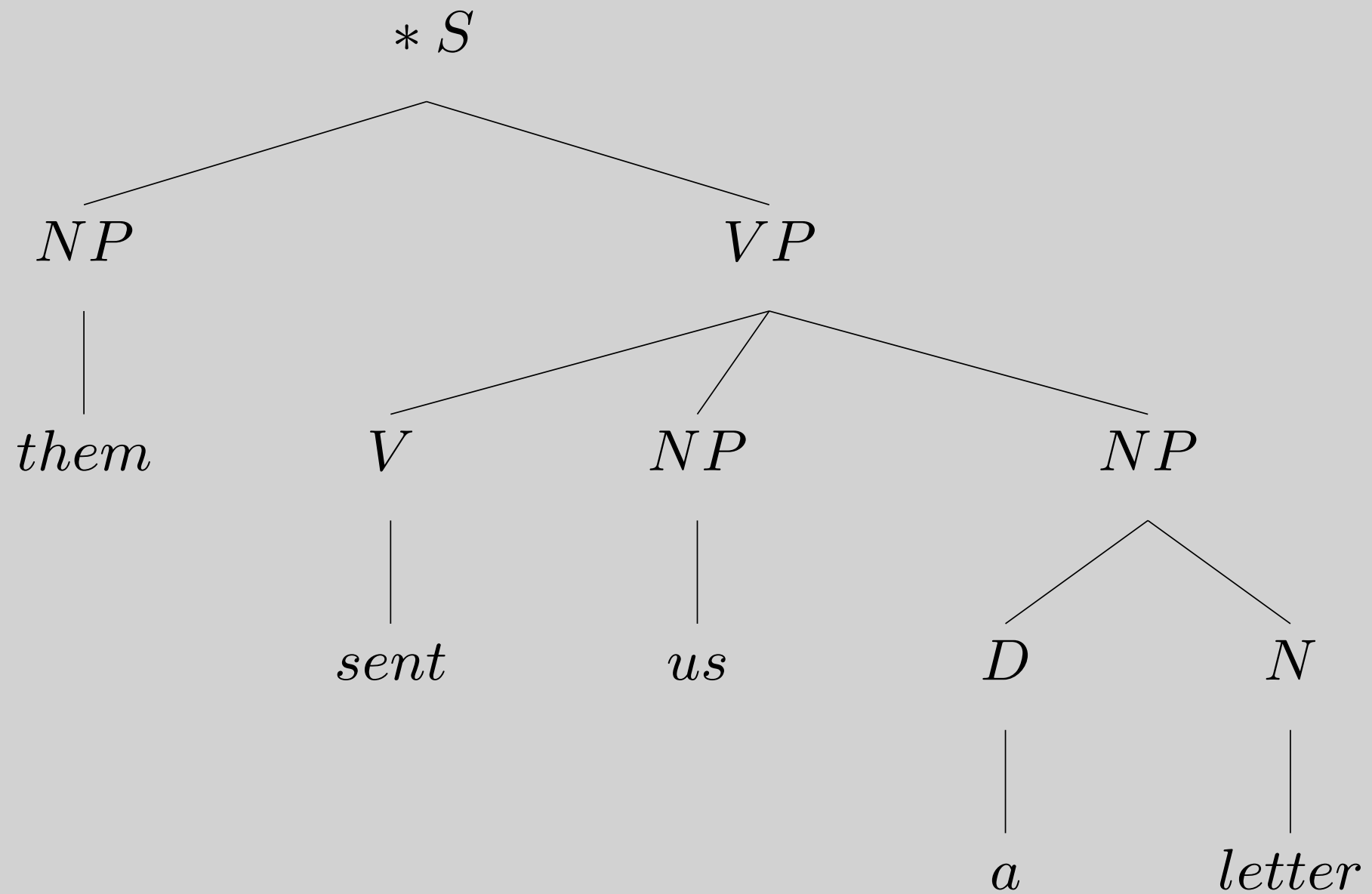
Two Semantic Features: the Lexicon & SIP



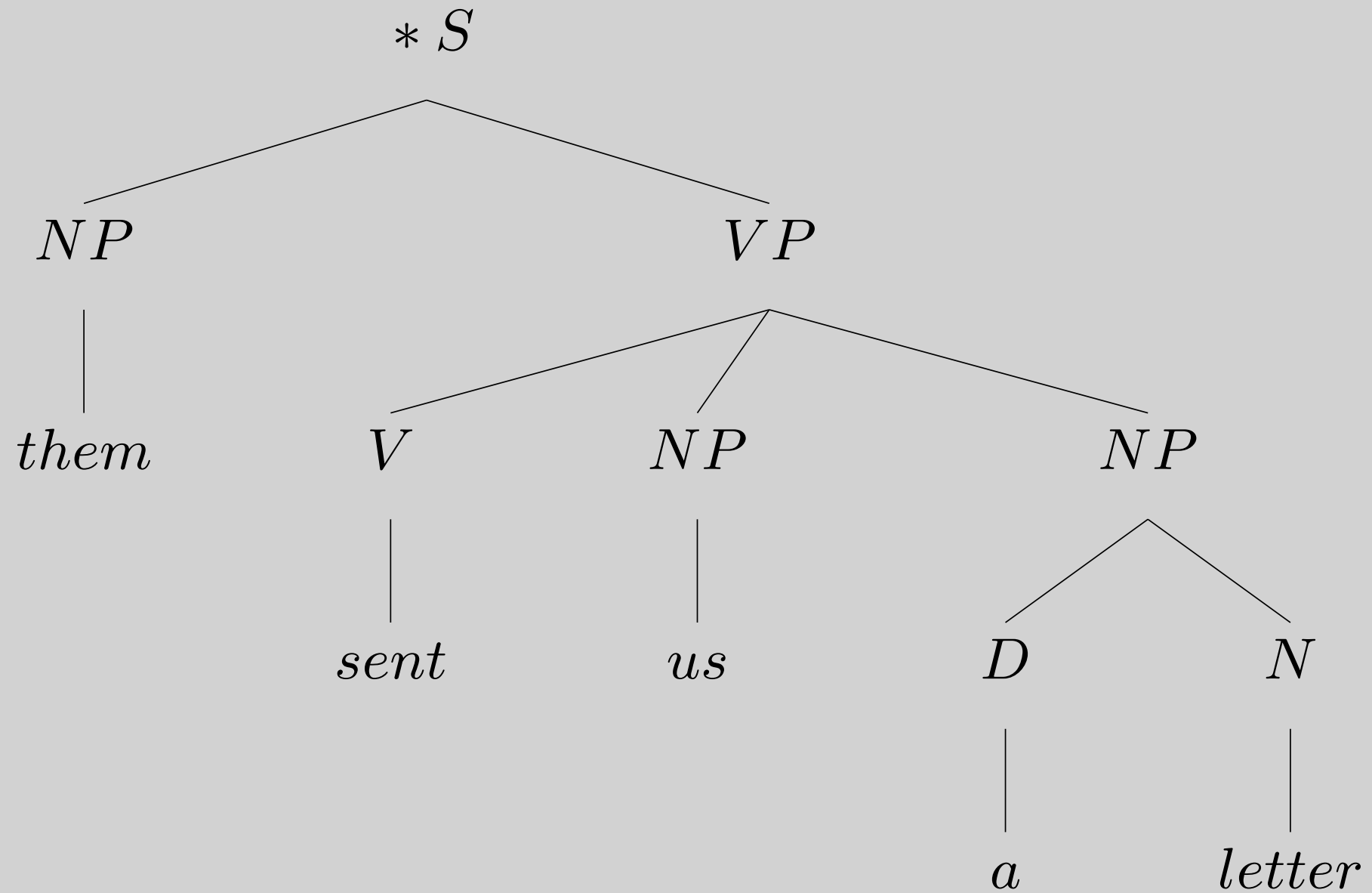
RESTR Values and the SCP



An Ungrammatical Example

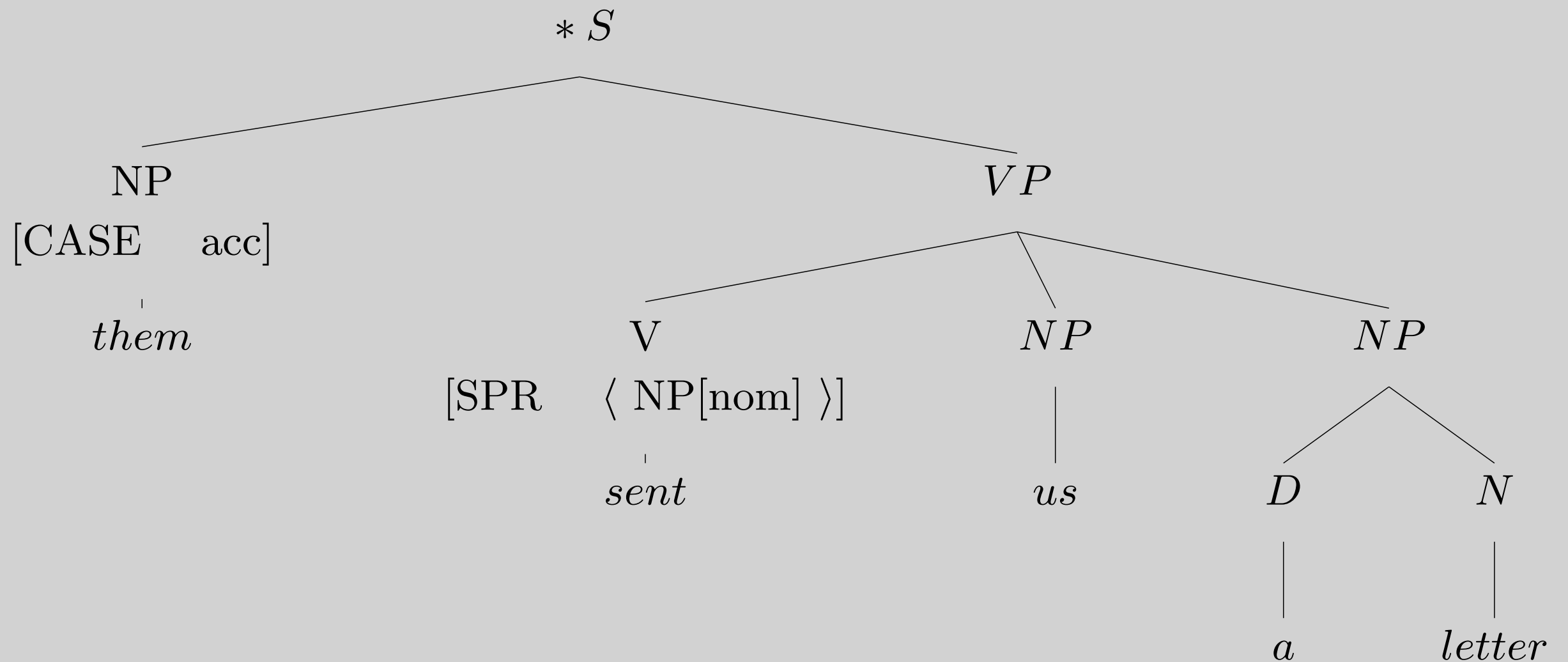


An Ungrammatical Example



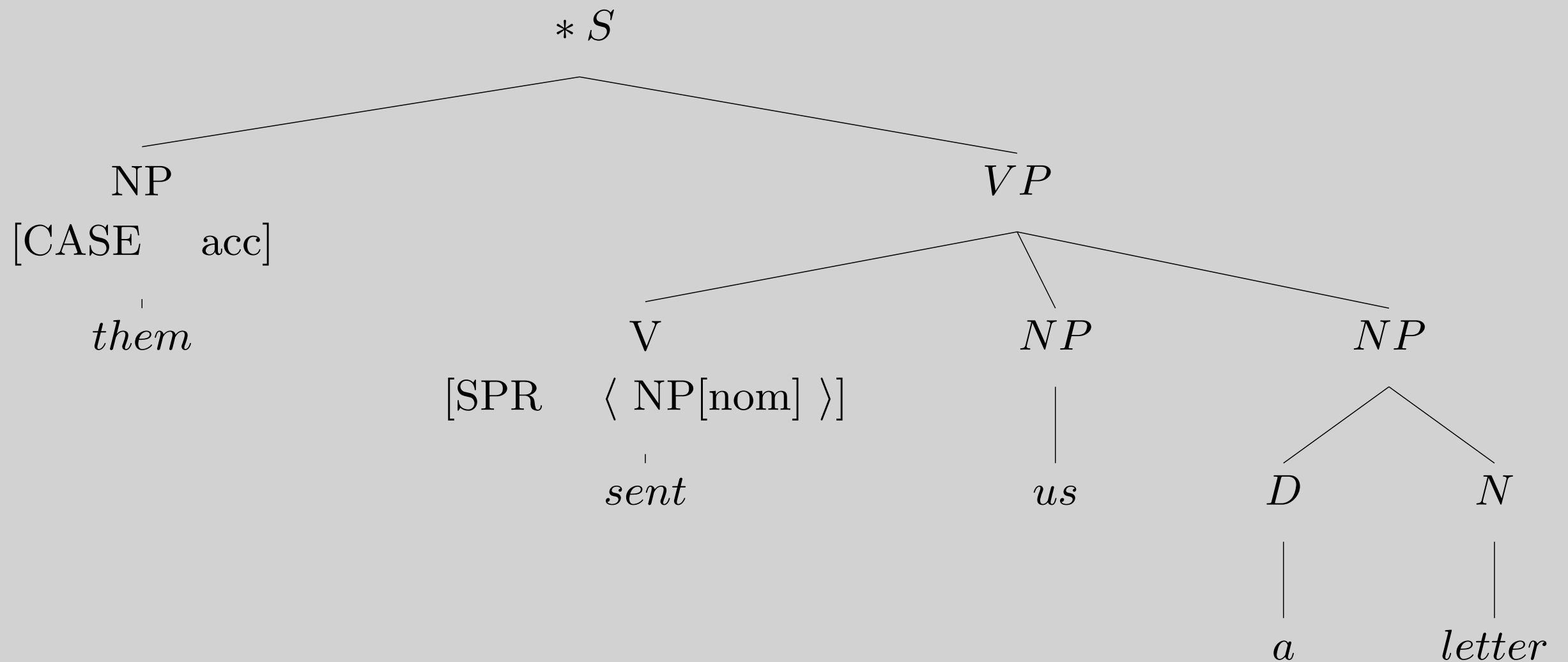
What's wrong with this sentence?

An Ungrammatical Example



What's wrong with this sentence?

An Ungrammatical Example

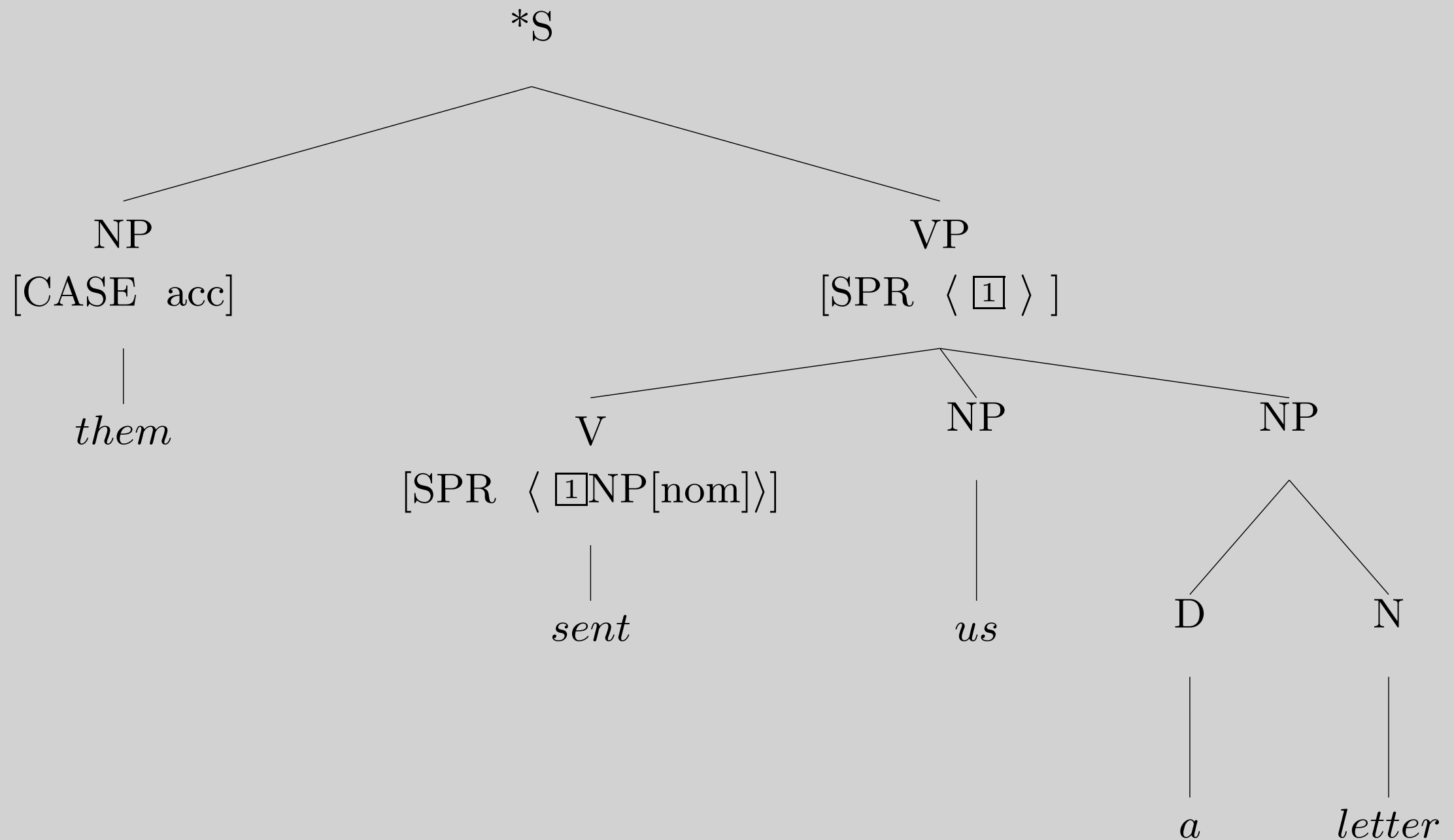


What's wrong with this sentence?

So what?

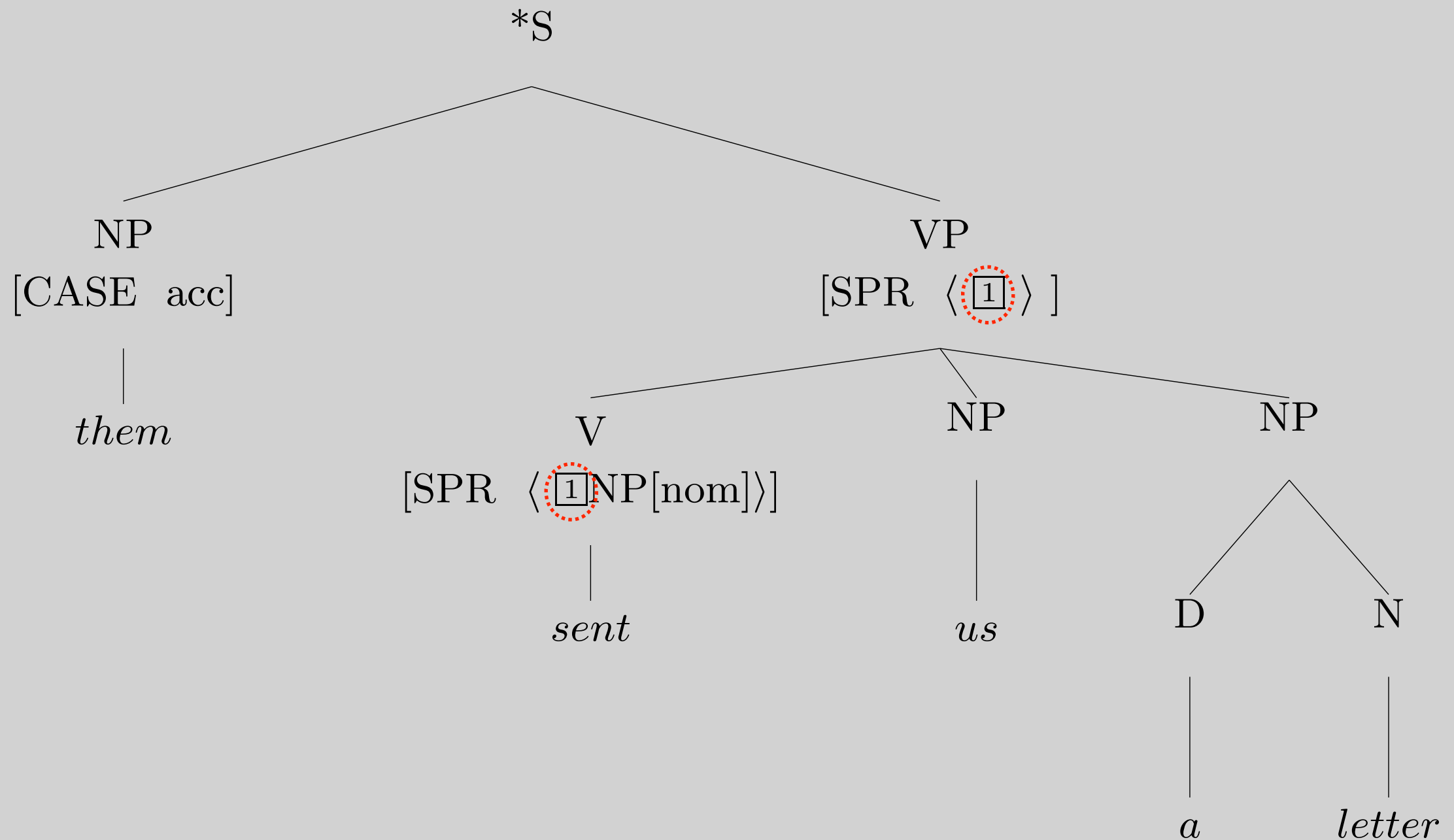
An Ungrammatical Example

The Valence Principle



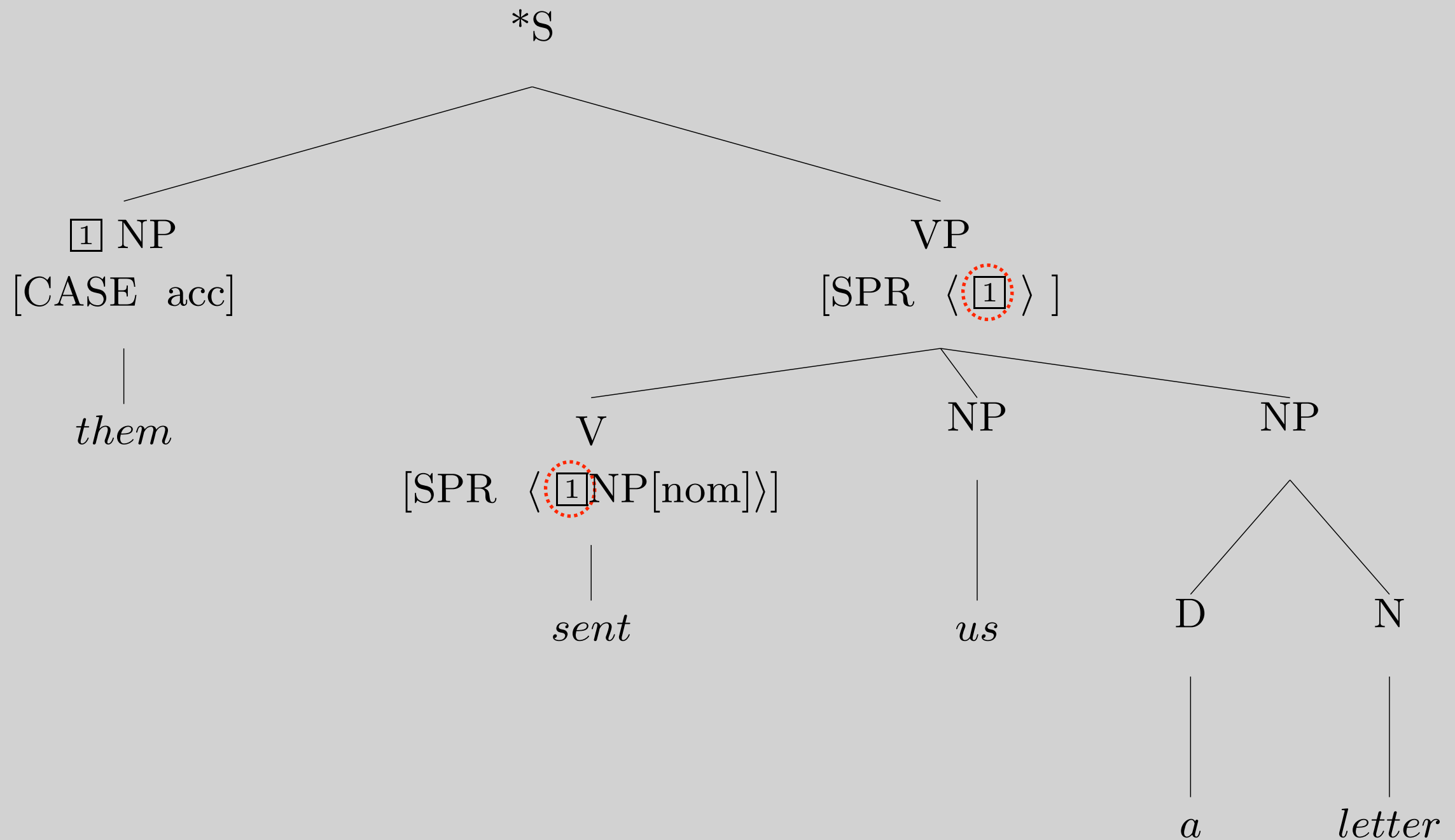
An Ungrammatical Example

The Valence Principle



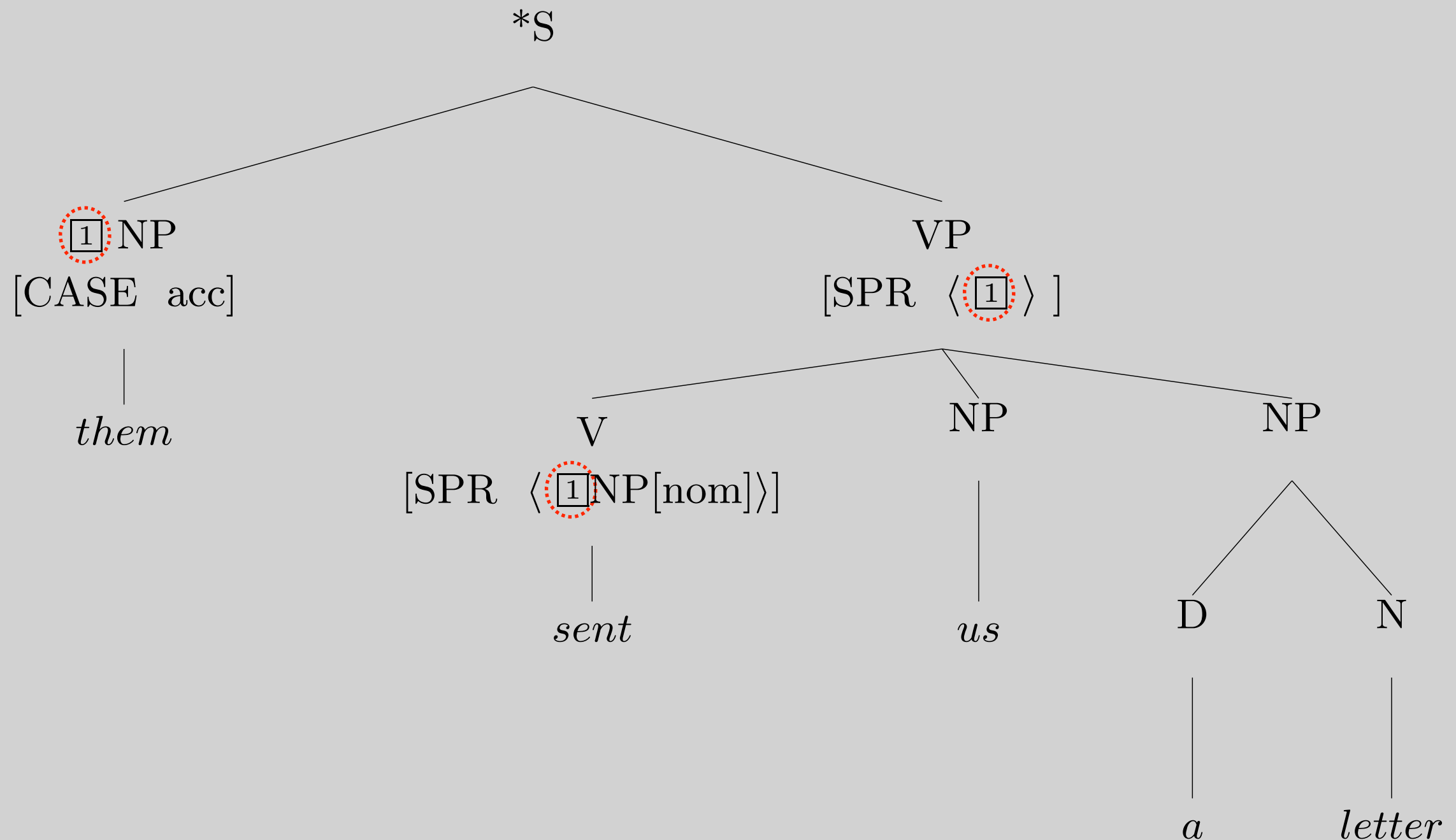
An Ungrammatical Example

Head Specifier Rule



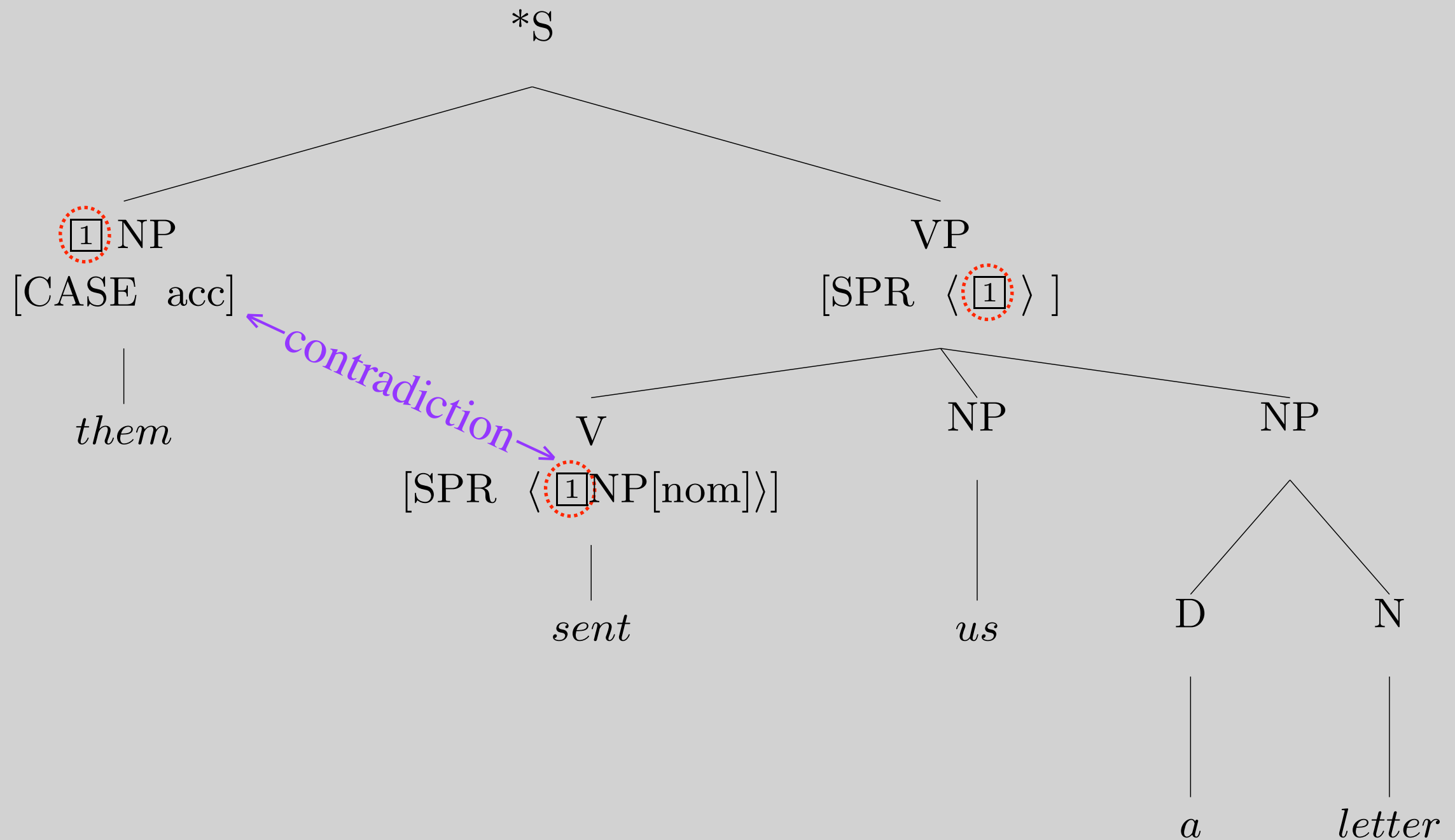
An Ungrammatical Example

Head Specifier Rule



An Ungrammatical Example

Head Specifier Rule



Exercise in Critical Thinking

Exercise in Critical Thinking

- Our grammar has come a long way since Ch 2, as we've added ways of representing different kinds of information:
 - generalizations across categories
 - semantics
 - particular linguistic phenomena: valence, agreement, modification

Exercise in Critical Thinking

- Our grammar has come a long way since Ch 2, as we've added ways of representing different kinds of information:
 - generalizations across categories
 - semantics
 - particular linguistic phenomena: valence, agreement, modification
- What else might we add? What facts about language are as yet unrepresented in our model?

Overview

- What we're trying to do
- The pieces of our grammar
- Two extended examples
- Reflection on what we've done, what we still have to do
- Reading questions
- Next time: Catch up & review

Reading Questions

- How do we know whether it's D taking the N as COMPS, or the N taking the D as SPR?
- More generally, how do we decide how many semantic & syntactic arguments to posit? Why does letter have an ADDRESSEE? What if there's more than one addressee?
- What is INST for?

Reading Questions

- Is a lexical entry like an abbreviation? If not, how do we decide how much info to put in them?
- How do you build a lexicon, practically speaking?
- Why do we want NP and N to (sometimes) be equivalent?