Ling 566
Oct 22, 2015

Review

Overview

- Reading questions
- Homework tips
- SPR and COMPS
- Common mistakes
- Analogies to other systems you might know

Reading Questions

- Do we have to understand 6.3 (the squiggly bits)?
- I am wondering what exactly ω and Φ stand for in 6.1. From the context, it looks like ω may stand for the surface word, whereas Φ stands for the specified features of a given interpretation of that word. 'F' is specified as a "resolved feature structure", but the other symbols do not have explicit definitions.

First, each lexical entry licenses a family of word structures – each of which is a nonbranching tree. More precisely, a lexical entry $\langle \omega, \Phi \rangle$ licenses any word structure of the form:

$$\begin{tikzcd}
\mathrm{F} \arrow[d, dash] \\
\omega
\end{tikzcd}$$

if and only if F is a resolved feature structure that satisfies Φ. A resolved feature structure F satisfies Φ if and only if it assigns values to all features appropriate for feature structures of its type, and those values are consistent with all of the information specified in Φ.

Reading Questions

- In the appendix it mentions that feature structures have a recursive definition. Why do they need to have a recursive definition and which part of the definition is recursive?
- What is the difference between sequences ϕ and description sequences d?

Reading Questions

- In 6.3.5, a requirement of a tree structure is: 3. sister nodes are ordered with respect to each other. Is this the same as saying there can on be only one possible ordering of nodes in a given structure?
- And another requirement is: 4. it has no crossing branches What's an example of a spurious structure that would have crossing branches?

Reading Questions

- From the examples in the chapter, it appears we can arbitrarily choose a gender value for word structures corresponding to proper nouns (names). How about cases when other words within the sentence (i.e. gender specific pronouns) give some indication of gender--would we then simply choose the gender based on that context?

Reading Questions

- Earlier in the class, we discussed how the book states that feature structures need to be fully resolved. In this chapter, though, example 8 states that the addressee field does not need to reference anything. Is it still a fully resolved tree, even if the addressee is not referencing anything? What's the difference between this case, and a case that would not accept a tree because it isn't fully resolved?

Reading Questions

- Because of the change on the morphology of the word, it makes sense why we have to create two separate lexical entries for the same verb based on the tense (*send* vs. *sent*). And it also makes sense why we have to make a case for agreement for the present tense of the verb (*send* vs. *sends*). However, for the past tense (*sent*), the word isn't morphologically affected when it is used with either 3rd, 2nd, 1st, plural or single NPs, thus it seems unnecessary to have to specify AGR for the verb *sent*.

Reading Questions

- The verb sent in example (13), the COMPS list includes two NPs both with [CASE acc]. I understand the CASE constraint on the first NP, but don't quite understand why the second NP also has a CASE constraint. At least in English, I haven't been able to think of an example using sent where the second NP would be a pronoun where CASE would be meaningful. In our example it is *a letter*.

- Why do we put CASE outside of AGR? (as in pg. 167 (2a))

Reading Questions

- Are prepositions always semantically empty? What about *This is for you*?

- (28) shows the phrase *to Lee*, and points out that the preposition *to* has no semantics on its own. I get the feeling that this isn't a necessary consequence of the grammar so far, but instead is something of a stylistic choice. Would it be straightforward to get the same semantics in the end, if prepositions like *to* have their own semantic interpretation?

Reading Questions

- I'd like to know how we define the meaning of the RELN values. It seemed like we made use of a times relation to crate two hundred from two and hundred. Yet we didn't explicitly define what that means. Is it just a place marker?

- I was a bit surprised to see RESTR values for "two" and "letters" that where called two and letter. Perhaps I shouldn't be -- since we obviously have to have some grasp of the language used in our formalisms (and it just so happens that it's the same language we're analyzing) and since all of the SEM structures up until now have involved English words -- but it nevertheless struck me as circular in these cases. Why is that seeming circularity not considered a problem for the grammar, especially when one gets to the point of trying to implement NLP?

Reading Questions

- The RESTR value of "us" contains three predications; send, group, and speaker. In the sentence "they sent us a letter" the INST of group is identified with the SENDEE feature of "send" but the other two predications don't show up again. So I was wondering what purpose those predications serve? Are there sentences where they are connected to other semantic entries?

Reading Questions

- Since there seem to be various different ways to define the SEM RESTR values how to you know when you have enough predications?
- On the phrase level, the RESTR value order appears to be determined by word order within the phrase. How does this apply to the word level? How do we know RESTR value predication order for a lexical entry?

Reading Questions

- We don't, however, actually know the specific identities of *they* or *us* without more context. Imagine the sentence, *They sent us a letter* occurred in the context, *My sister and I emailed our grandparents*. *They sent us a letter.* Could we use the indices already described to connect *my sister and I* with *us* and *our grandparents* with *they*? Perhaps we could extrapolate the Semantic Compositionality Principle to a wider scope? This seems related to issues like anaphora resolution.

Reading Questions

- In a sentence, it seems that the RESTR value of the verb is a good indicator of how many semantic indices there will be. However, I'm not 100 % certain how to annotate more complicated NP's which contain NP's such as *Jake's friend* or *the cat on the mat in the house*. It seems that the Semantic Inheritance principle would reduce each of those NP's into a single index as in *two letters to Lee* on page 190; this would lead me to believe that every noun should have its own index.

Reading Questions

- In languages that use complex case systems, it seems to me that there would be certain overlap between semantic and syntactic features. How could redundancy be avoided (or should it be)?

Reading Questions

- Which is used more frequently in real-life computational linguistics, and what are the qualities that might make one sentence more amenable to a given methodology?

- In the book, I felt that for the top down approach, a list of RESTR predications are immediately introduced, but is there a good technique / approach / advice on how to come up with such predications at the first step? It just seems counter-intuitive to do it this way because it feels like a process of dividing up the list of RESTR, instead of summing up the RESTR.

Reading Questions

- It says the top-down approach could be used equally well, but in the example, starts immediately with RESTR lists that only could have been generated with a human understanding of the sentence, and tree that is already constructed. I understand that trees can be analyzed top-down and rules can be applied to license its parts from the top-down, but I don't understand how the tree could actually be constructed from the top down. (Or, if it can be done more intelligently than brute force, what reason there would be to do so.)

Reading Questions

- Could top-down and bottom-up parsing be combined (in computational applications) in an effort to disambiguate structural/word sense/etc ambiguity? There would obviously need to be some probabilistic weights involved from both ends.

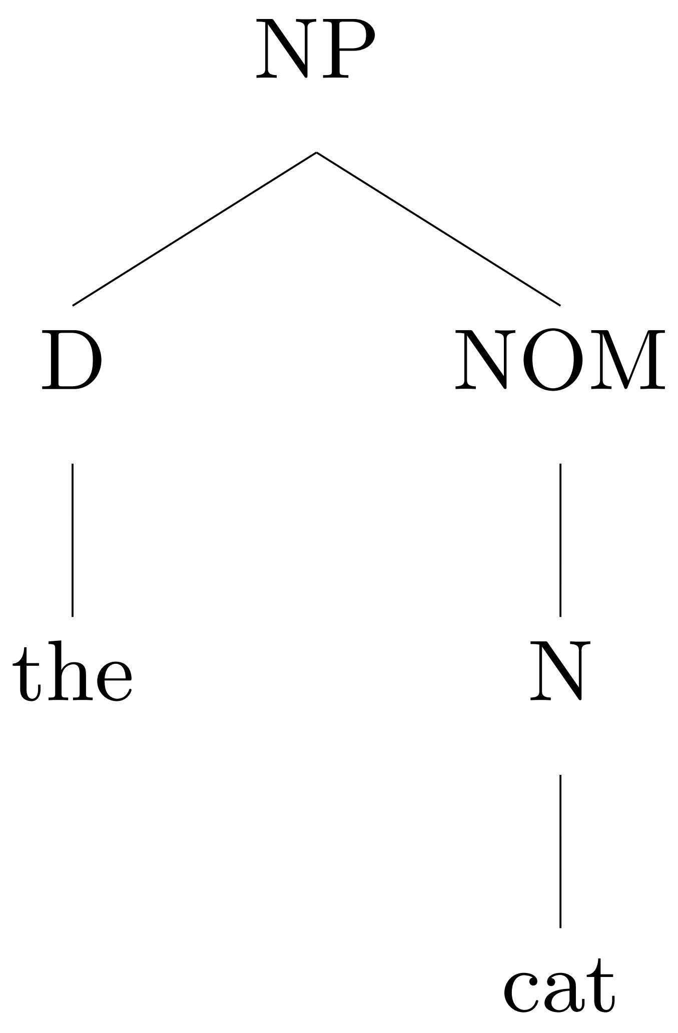
Homework tips/requests

- Type whenever possible
- Answer each part of each question separately
- Be sure to answer each part of each question, and follow the directions!
- Look over the problems early and ask questions
- Check your work
- Monitor GoPost
- WORK TOGETHER

SPR value on AP/PP?

- Kim grew fond of baseball.
- Kim and Sandy ate lunch in the park.
- Kim and Sandy are in the park.

Which grammar does this tree go with?



What’s wrong with this?

$$\left\langle \text{out}, \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{prep} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \text{VP}\ \rangle \\ \text{COMPS} & \langle\ (\text{PP} \mid \text{NP})\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

What's wrong with this?

$$\left\langle \text{out}, \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{prep} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ (\text{NP})\ (\text{PP})\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

What’s wrong with this?

$$\left\langle \text{out}, \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{prep} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & (\ \text{NP} \mid \text{PP}\ ) \end{bmatrix} \end{bmatrix} \right\rangle$$

What's wrong with this?

$$\left\langle \text{grew}, \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \textit{3sing} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \text{ NP } \rangle \\ \text{COMPS} & \langle \text{ AP } \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

What's wrong with this?

$$\left\langle \text{out}, \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{preposition} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\,\rangle \\ \text{COMPS} & \langle\ (\ \text{NP} \mid \text{PP}\ )\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

What's wrong with this?

$$\left\langle \text{there}, \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{prep} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Tags & lists

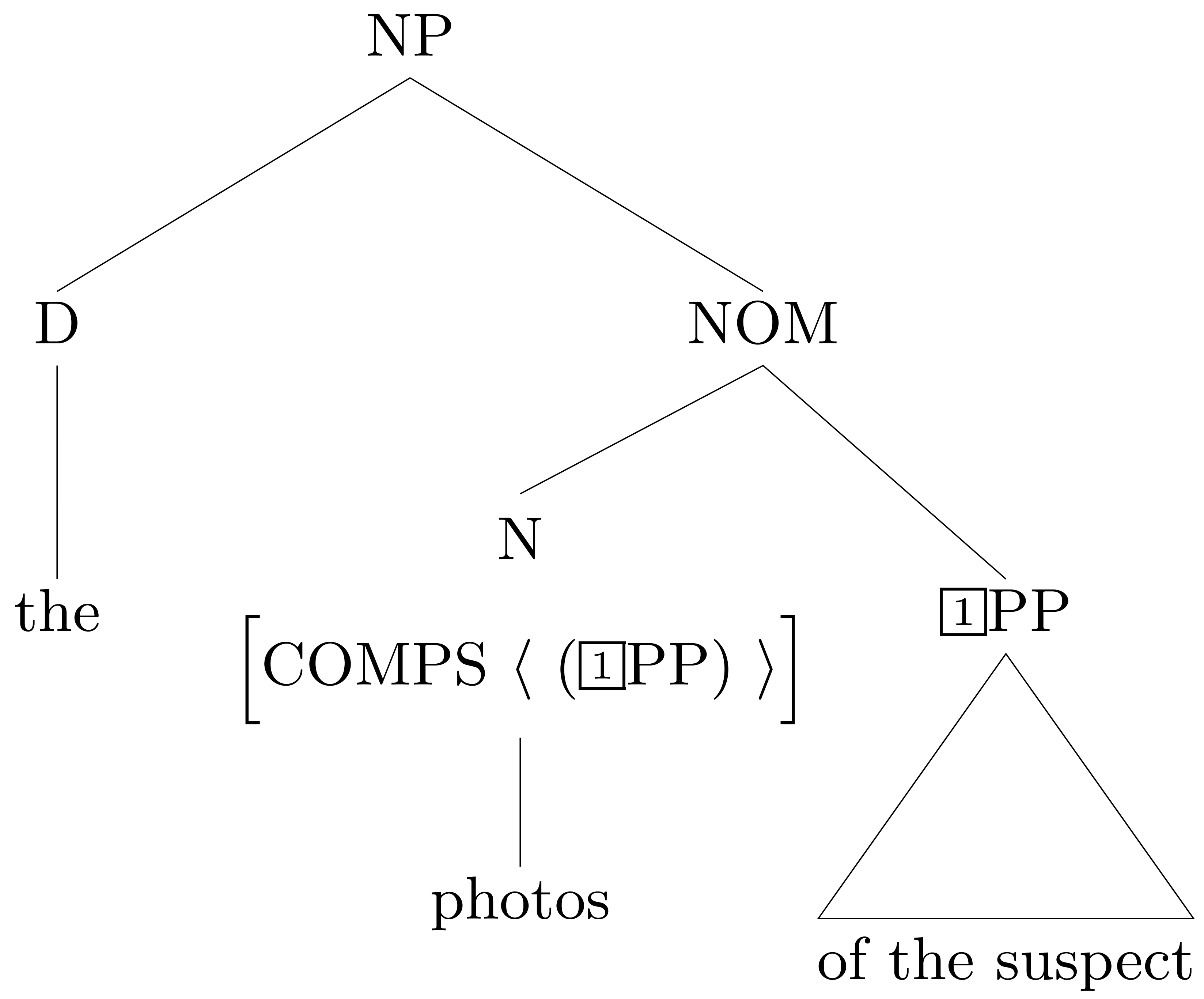
- What's the difference between these two?

$$\begin{bmatrix}\text{SPR} & \boxed{1}\langle\ \text{NP}\ \rangle\end{bmatrix}$$

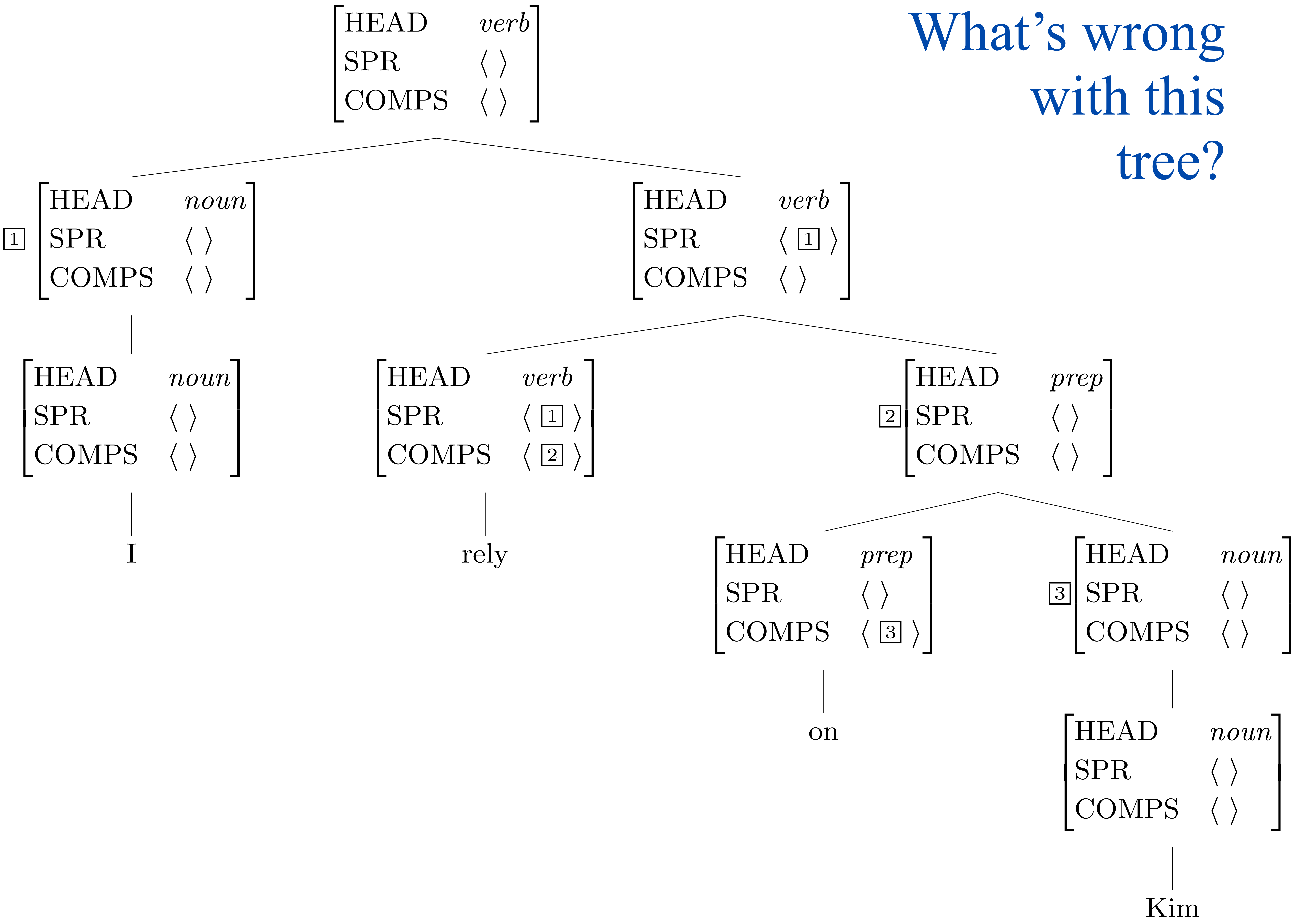
$$\begin{bmatrix}\text{SPR} & \langle\ \boxed{1}\text{NP}\ \rangle\end{bmatrix}$$

- When does it matter?

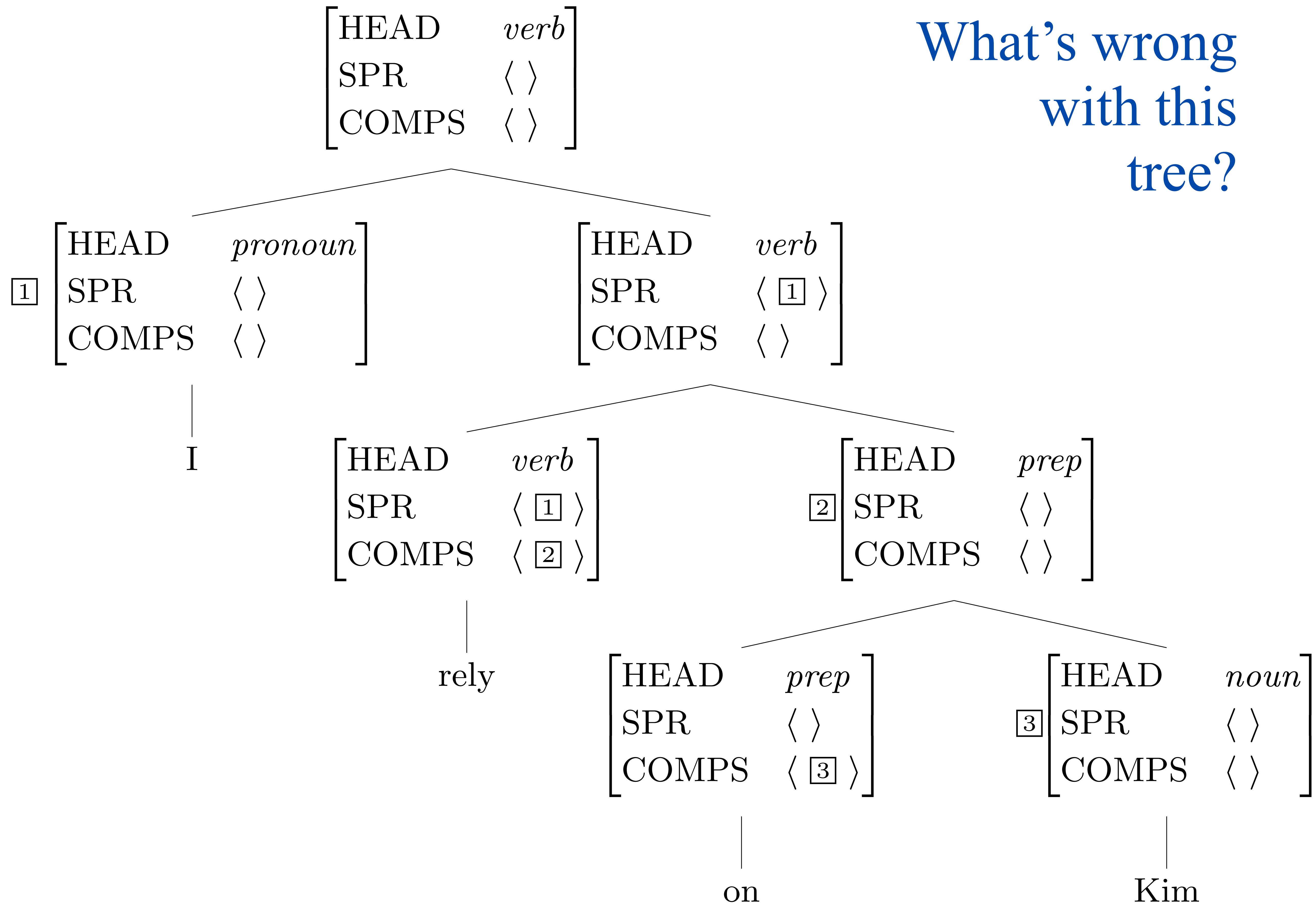
What’s wrong with this tree?


NP
D
NOM
the
N
$\left[\text{COMPS } \langle\ (\boxed{1}\text{PP})\ \rangle\right]$
photos
$\boxed{1}$PP
of the suspect

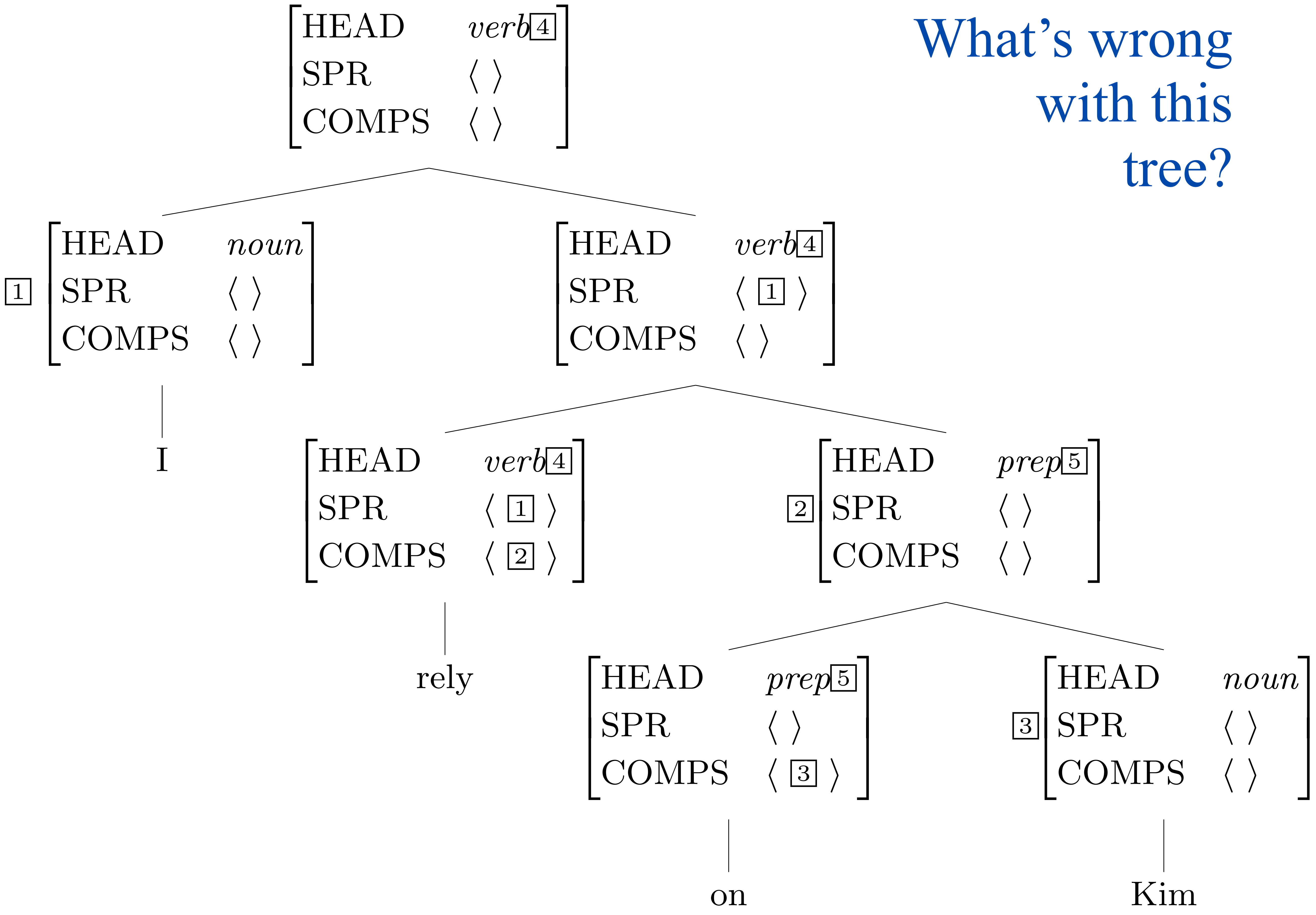
What's wrong with this tree?



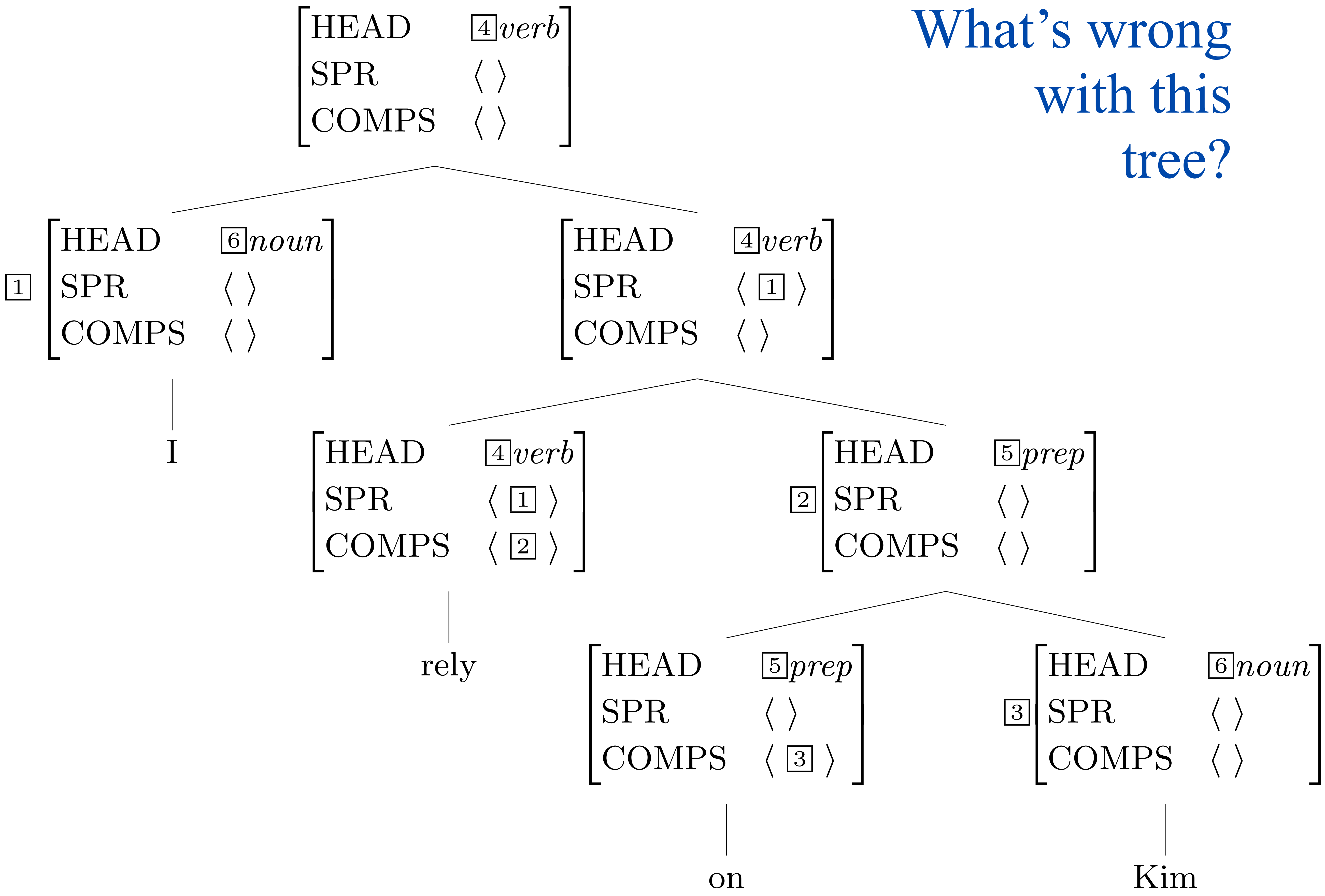
What's wrong with this tree?



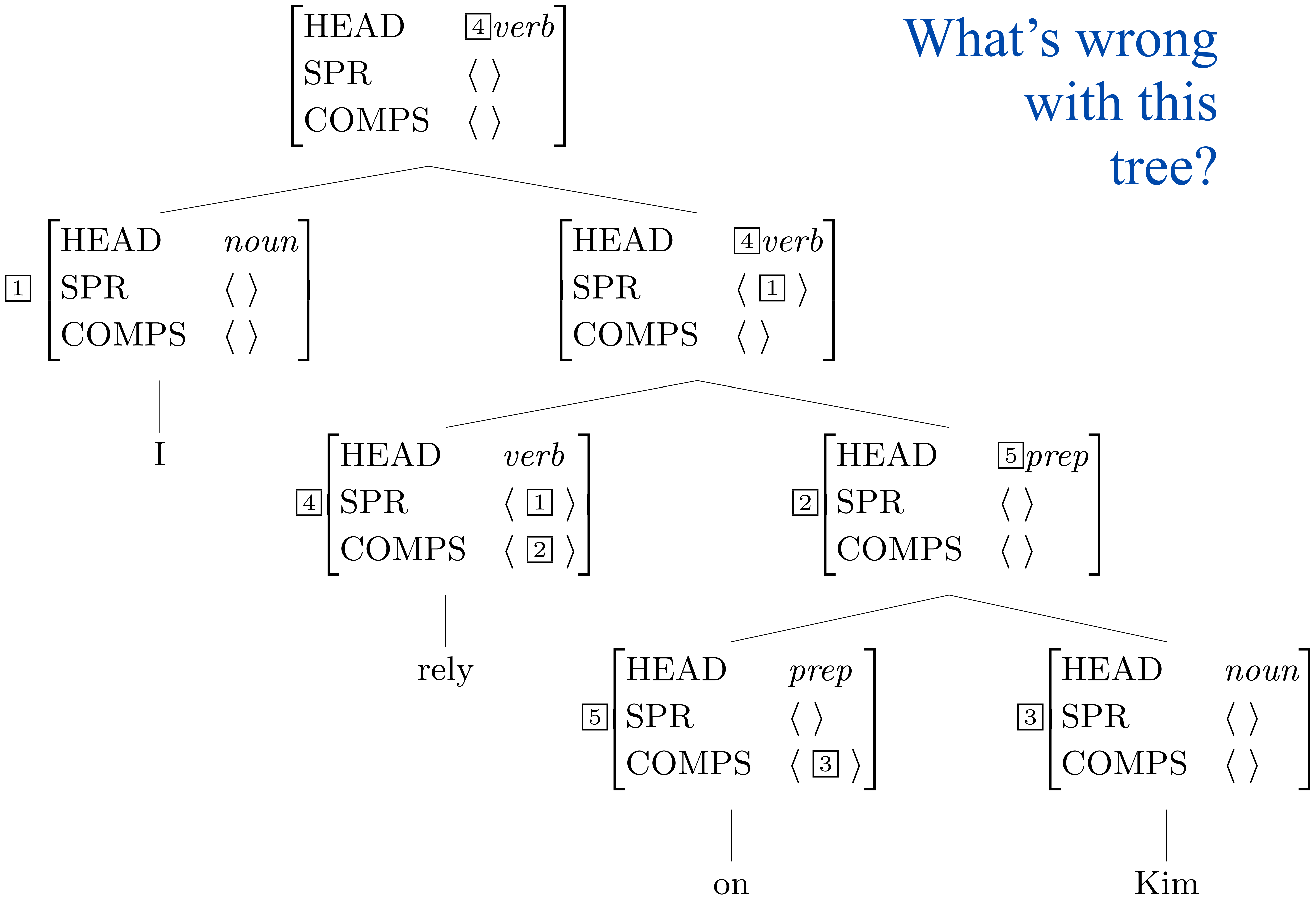
What's wrong with this tree?



What's wrong with this tree?



What's wrong with this tree?



What's wrong with this?

$$\left\langle \text{hundred}\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{number} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{bmatrix} \text{HEAD} & \textit{number} \\ \text{INDEX} & j \end{bmatrix} \right\rangle \\ \text{COMPS} & \left\langle \begin{bmatrix} \text{HEAD} & \textit{number} \\ \text{INDEX} & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & m \\ \text{MODE} & \textit{ref} \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textit{hund-number} \\ \text{MULTIPLIER} & j \\ \text{ADDEND} & k \\ \text{HUND-VALUE} & m \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

And this?

$$\left\langle \text{hundred}\ , \left[\begin{array}{ll} \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \textit{number} \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle \left[\text{HEAD} \quad \textit{number}\right] \rangle \\ \text{COMPS} & \langle \left[\text{HEAD} \quad \textit{number}\right] \rangle \end{array}\right] \end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll} \text{INDEX} & i \\ \text{MODE} & \textit{ref} \\ \text{RESTR} & \left\langle \begin{array}{l} \left[\begin{array}{ll} \text{RELN} & \textit{times} \\ \text{RESULT} & k \\ \text{FACTOR1} & l \\ \text{FACTOR2} & m \end{array}\right] , \left[\begin{array}{ll} \text{RELN} & \textit{constant} \\ \text{INST} & m \\ \text{VALUE} & \textit{100} \end{array}\right] , \\ \left[\begin{array}{ll} \text{RELN} & \textit{plus} \\ \text{RESULT} & i \\ \text{TERM1} & j \\ \text{TERM2} & k \end{array}\right] \end{array} \right\rangle \end{array}\right] \end{array}\right] \right\rangle$$

How about this?

$$\left\langle \text{hundred}\,,\ \left[\begin{array}{ll}
\text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \textit{number} \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \left\langle \left[\begin{array}{ll}\text{HEAD} & \textit{number}\\ \text{INDEX} & l\end{array}\right]\right\rangle \\ \text{COMPS} & \left\langle \left[\begin{array}{ll}\text{HEAD} & \textit{number}\\ \text{INDEX} & j\end{array}\right]\right\rangle \end{array}\right]\end{array}\right] \\
\text{SEM} & \left[\begin{array}{ll} \text{INDEX} & m \\ \text{MODE} & \textit{ref} \\ \text{RESTR} & \left\langle \begin{array}{l} \left[\begin{array}{ll}\text{RELN} & \textit{times}\\ \text{RESULT} & k \\ \text{FACTOR1} & l \\ \text{FACTOR2} & m\end{array}\right], \left[\begin{array}{ll}\text{RELN} & \textit{constant}\\ \text{INST} & m\\ \text{VALUE} & \textit{100}\end{array}\right], \\ \left[\begin{array}{ll}\text{RELN} & \textit{plus}\\ \text{RESULT} & i\\ \text{TERM1} & j\\ \text{TERM2} & k\end{array}\right]\end{array}\right\rangle \end{array}\right]
\end{array}\right]\right\rangle$$

Better version

$$\left\langle \text{hundred}\ , \begin{bmatrix} \mathrm{SYN} & \begin{bmatrix} \mathrm{HEAD} & \textit{number} \\ \mathrm{VAL} & \begin{bmatrix} \mathrm{SPR} & \left\langle \begin{bmatrix} \mathrm{HEAD} & \textit{number} \\ \mathrm{INDEX} & l \end{bmatrix} \right\rangle \\ \mathrm{COMPS} & \left\langle \begin{bmatrix} \mathrm{HEAD} & \textit{number} \\ \mathrm{INDEX} & j \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \\ \mathrm{SEM} & \begin{bmatrix} \mathrm{INDEX} & i \\ \mathrm{MODE} & \textit{ref} \\ \mathrm{RESTR} & \left\langle \begin{array}{l} \begin{bmatrix} \mathrm{RELN} & \textit{times} \\ \mathrm{RESULT} & k \\ \mathrm{FACTOR1} & l \\ \mathrm{FACTOR2} & m \end{bmatrix}, \begin{bmatrix} \mathrm{RELN} & \textit{constant} \\ \mathrm{INST} & m \\ \mathrm{VALUE} & \textit{100} \end{bmatrix}, \\ \begin{bmatrix} \mathrm{RELN} & \textit{plus} \\ \mathrm{RESULT} & i \\ \mathrm{TERM1} & j \\ \mathrm{TERM2} & k \end{bmatrix} \end{array} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Type hierarchy analogies

Type hierarchy analogies

- How is this formalism like OOP?

Type hierarchy analogies

- How is this formalism like OOP?
- How is it different?

Type hierarchy analogies

- How is this formalism like OOP?
- How is it different?
- How is the type hierarchy like an ontology?

Type hierarchy analogies

- How is this formalism like OOP?
- How is it different?
- How is the type hierarchy like an ontology?
- How is it different?

Type hierarchy analogies

- How is this formalism like OOP?
- How is it different?
- How is the type hierarchy like an ontology?
- How is it different?
- How is this formalism like the MP's formalism?

Type hierarchy analogies

- How is this formalism like OOP?
- How is it different?
- How is the type hierarchy like an ontology?
- How is it different?
- How is this formalism like the MP's formalism?
- How is it different?

Overview

- Reading questions
- Homework tips
- Common mistakes
- Analogies to other systems you might know