Syntactic Attributes

Morphology, heads, gaps, etc.

Note: The properties of nonterminal symbols are often called "features."
However, we will use the alternative name "attributes."

(We'll use "features" to refer only to the features that get weights in a machine learning model, e.g., a log-linear model.)

3 views of a context-free rule

- generation (production): $S \rightarrow NP\ VP$
- parsing (comprehension): $S \leftarrow NP\ VP$
- verification (checking): S = NP VP

- Today you should keep the third, declarative perspective in mind.

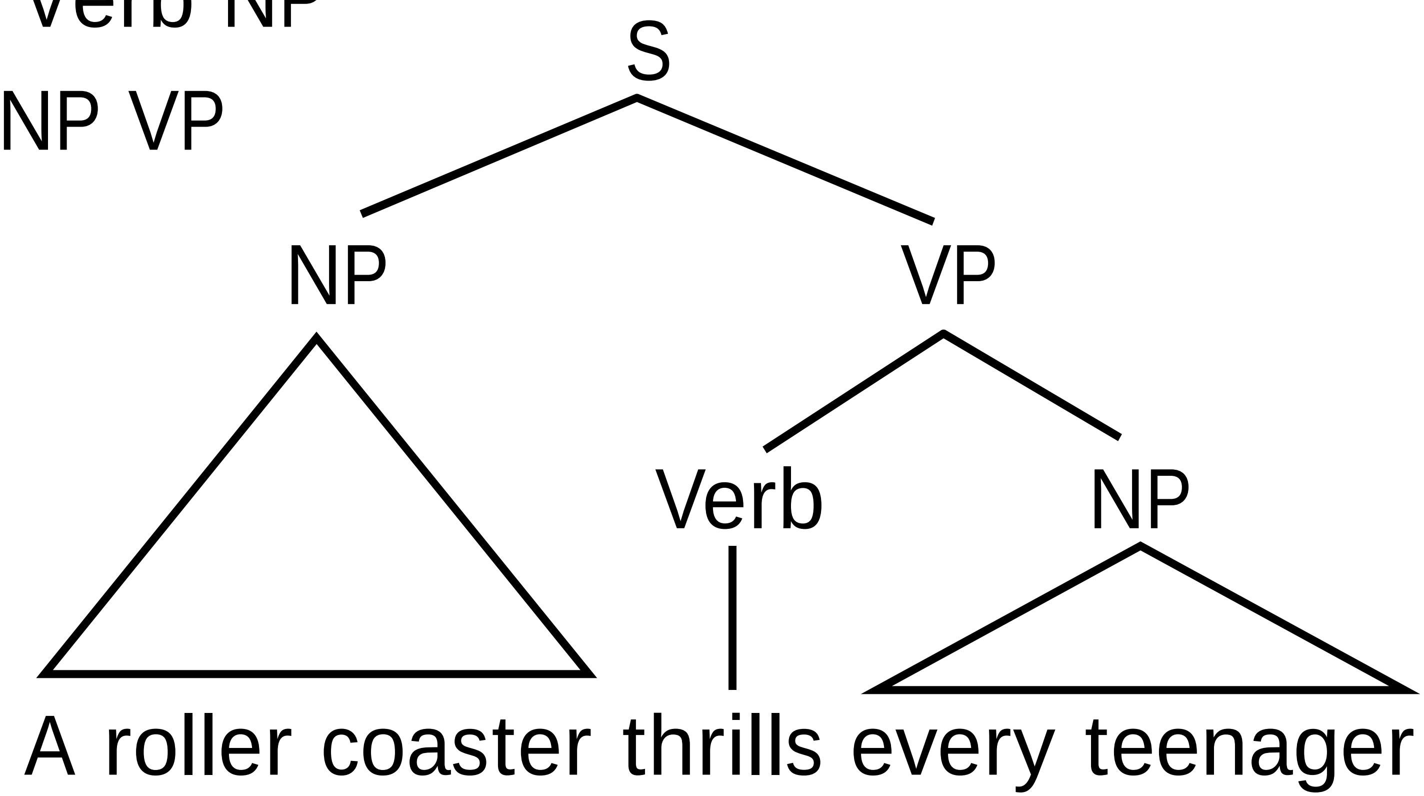
- Each phrase has
  - an interface (S) saying where it can go
  - an implementation (NP VP) saying what's in it
- To let the parts of the tree coordinate more closely with one another, enrich the interfaces: S[attributes...] = NP[attributes...] VP[attributes...]

Examples

Verb → thrills

VP→ Verb NP

S → NP VP



3 Common Ways to Use Attributes

morphology of a single word:

Verb[head=thrill, tense=present, num=sing, person=3,...] $\rightarrow$ thrills

projection of attributes up to a bigger phrase

VP[head=α, tense=β, num=γ...] $\rightarrow$ V[head=α, tense=β, num=γ...] NP

provided α is in the set TRANSITIVE-VERBS

agreement between sister phrases:

S[head=α, tense=β] $\rightarrow$ NP[num=γ,...] VP[head=α, tense=β, num=γ...]

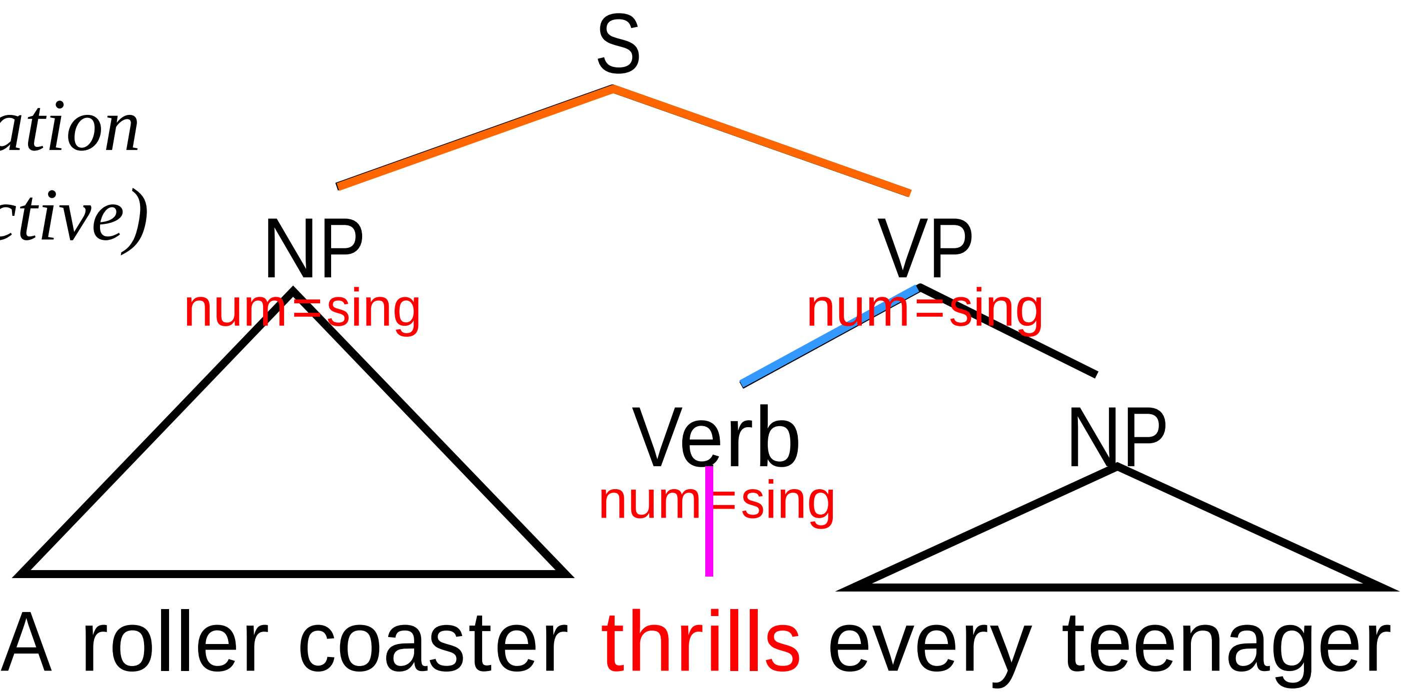
3 Common Ways to Use Attributes

Verb[head=thrill, tense=present, $\boxed{\text{num=sing}}$, person=3,...] → $\boxed{\text{thrills}}$

VP[head=α, tense=β, $\boxed{\text{num=}\gamma}$...] → V[head=$\alpha$, tense=$\beta$, $\boxed{\text{num=}\gamma}$...] NP

S[head=α, tense=β] → NP[$\boxed{\text{num=}\gamma}$,...] VP[head=$\alpha$, tense=$\beta$, $\boxed{\text{num=}\gamma}$...]

(generation perspective)



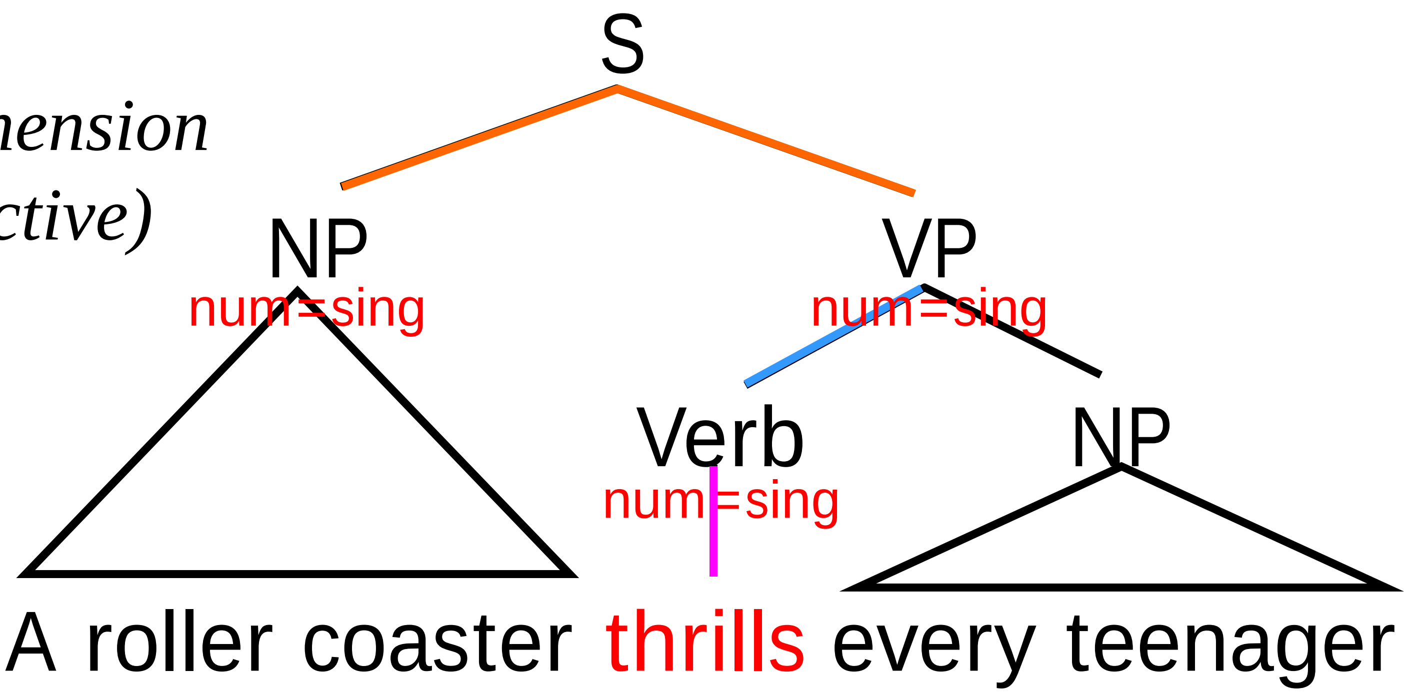
3 Common Ways to Use Attributes

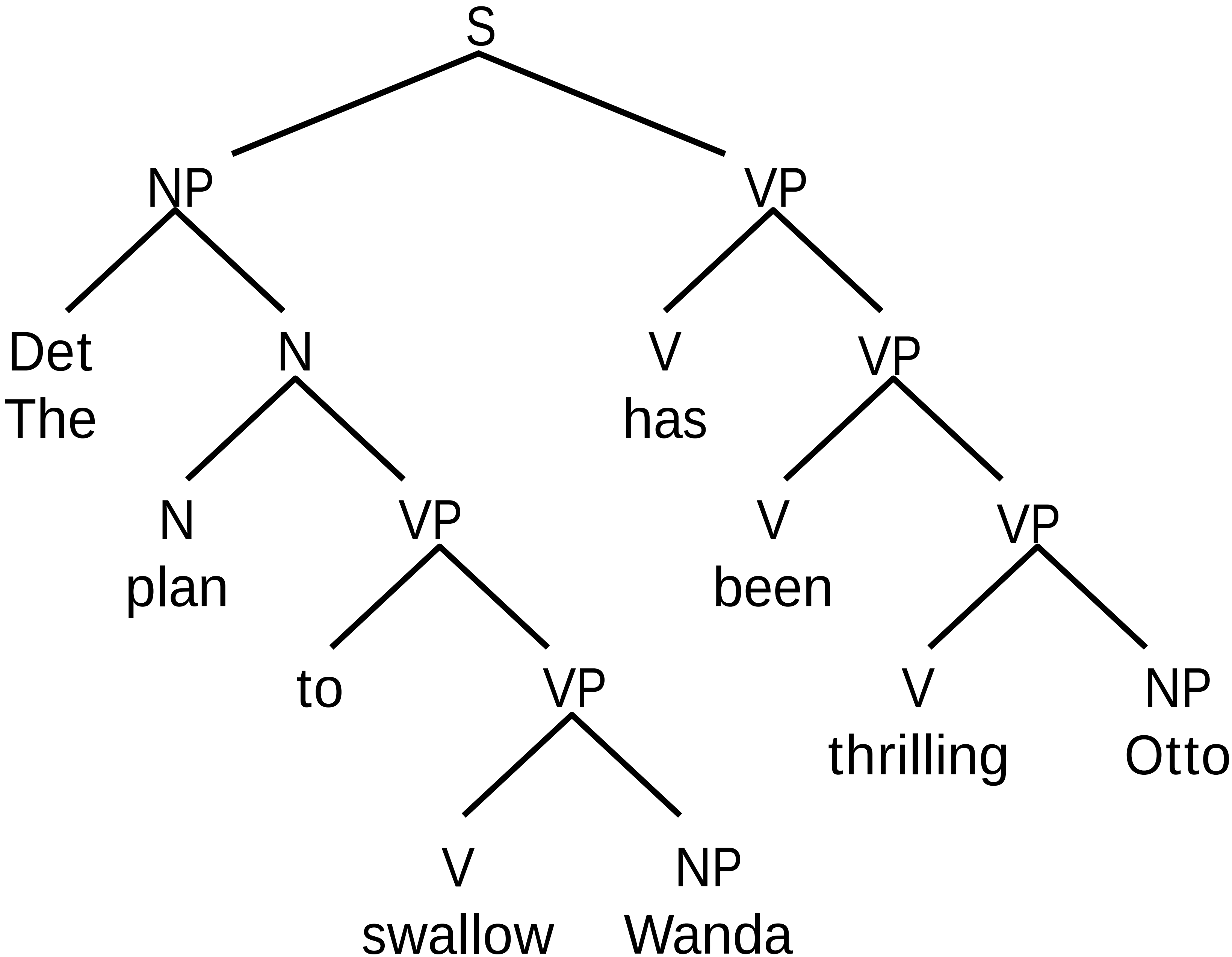
Verb[head=thrill, tense=present, $\boxed{\text{num=sing}}$, person=3,...] $\rightarrow$ $\boxed{\text{thrills}}$

VP[head=α, tense=β, $\boxed{\text{num=}\gamma}$...] $\rightarrow$ V[head=α, tense=β, $\boxed{\text{num=}\gamma}$...] NP

S[head=α, tense=β] $\rightarrow$ NP[$\boxed{\text{num=}\gamma\text{,}}$...] VP[head=$\alpha$, tense=$\beta$, $\boxed{\text{num=}\gamma}$...]

(comprehension perspective)

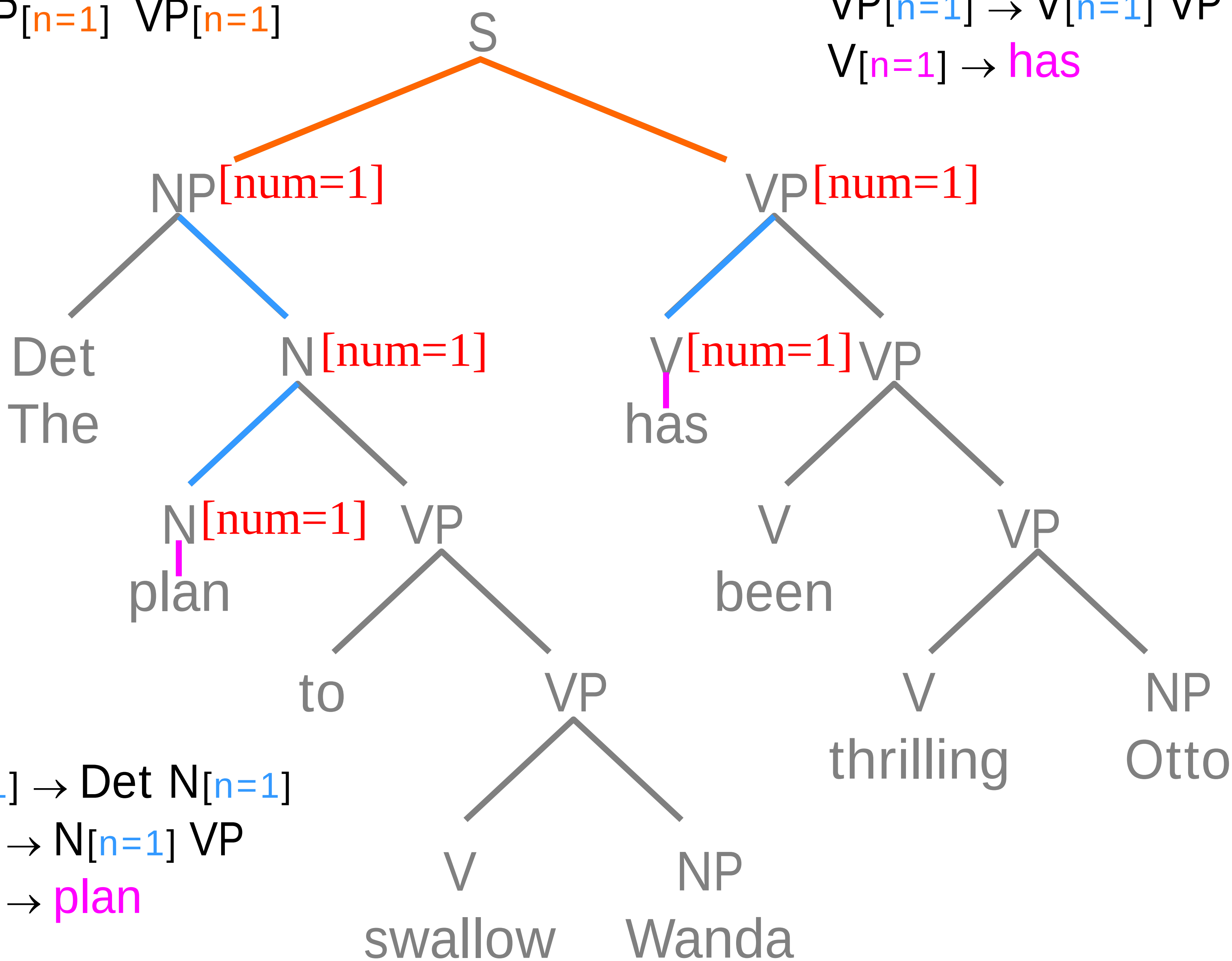



S
NP
VP
Det
The
N
V
has
VP
N
plan
VP
V
been
VP
to
VP
V
thrilling
NP
Otto
V
swallow
NP
Wanda

S→ NP[n=1] VP[n=1]

VP[n=1] → V[n=1] VP

V[n=1] → has



NP[n=1] → Det N[n=1]

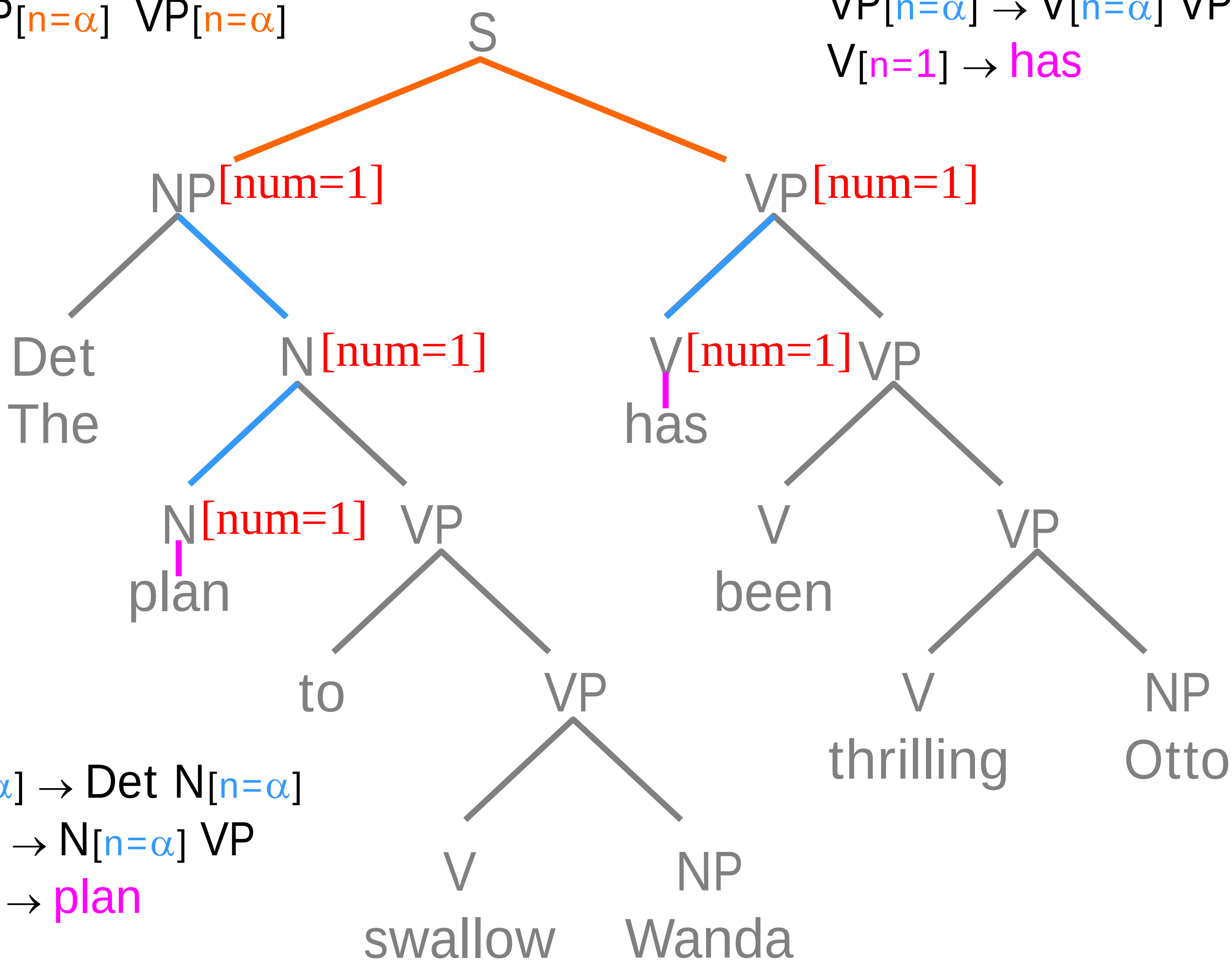
N[n=1] → N[n=1] VP

N[n=1] → plan

$S \rightarrow NP[n=\alpha]\ \ VP[n=\alpha]$

$VP[n=\alpha] \rightarrow V[n=\alpha]\ VP$

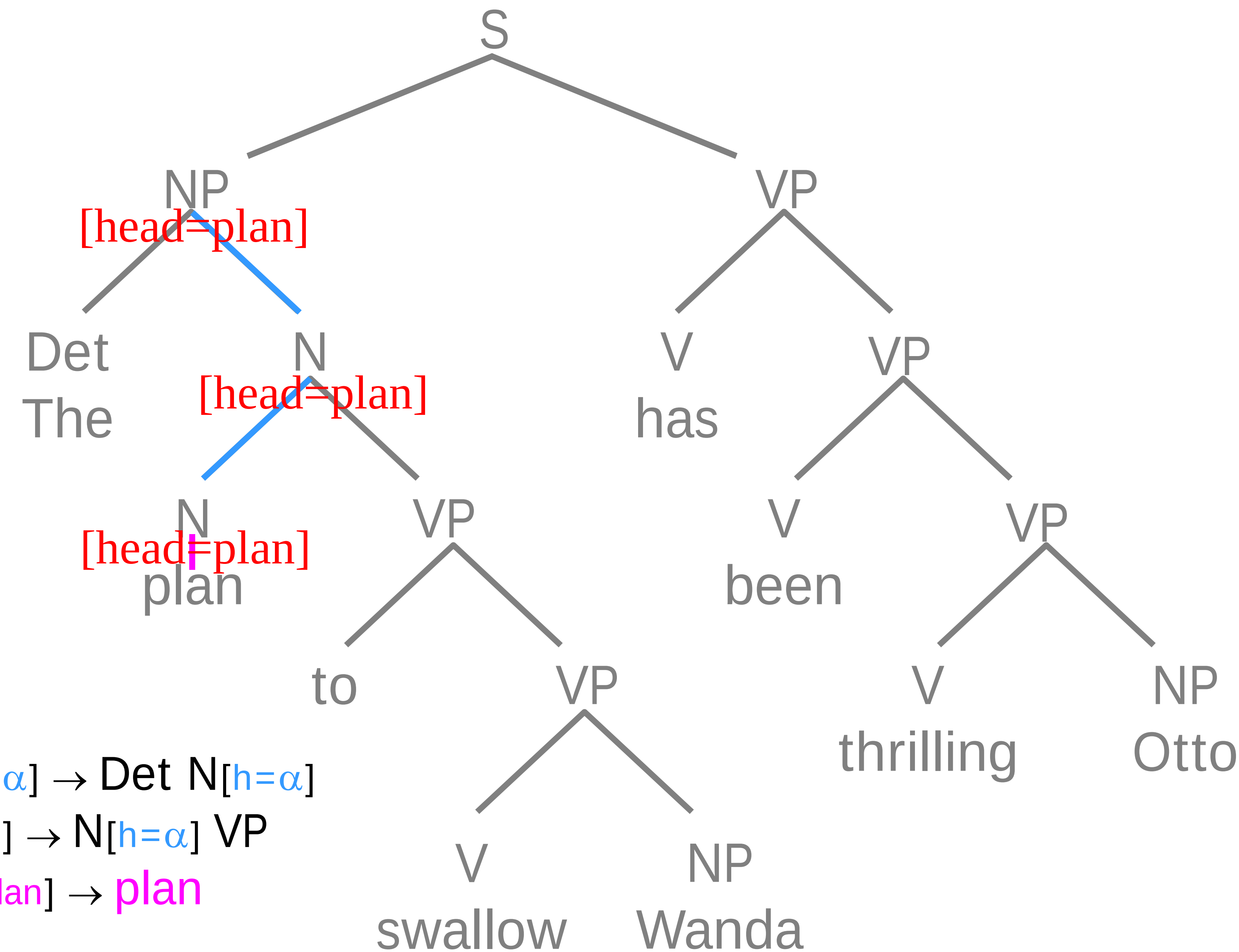
$V[n=1] \rightarrow$ has



$NP[n=\alpha] \rightarrow Det\ N[n=\alpha]$

$N[n=\alpha] \rightarrow N[n=\alpha]\ VP$

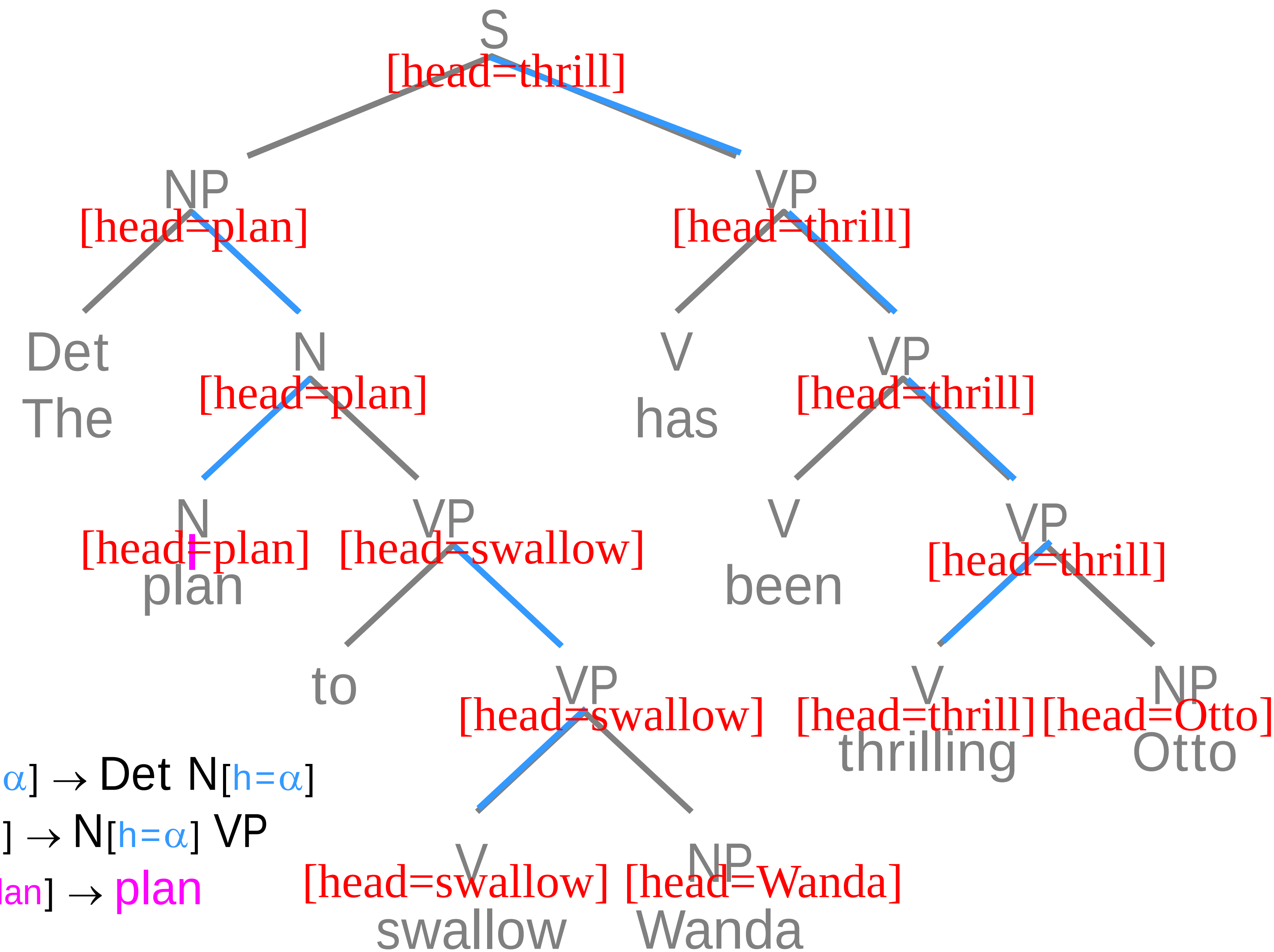
$N[n=1] \rightarrow$ plan



NP[h=α] $\rightarrow$ Det N[h=α]

N[h=α] $\rightarrow$ N[h=α] VP

N[h=plan] $\rightarrow$ plan

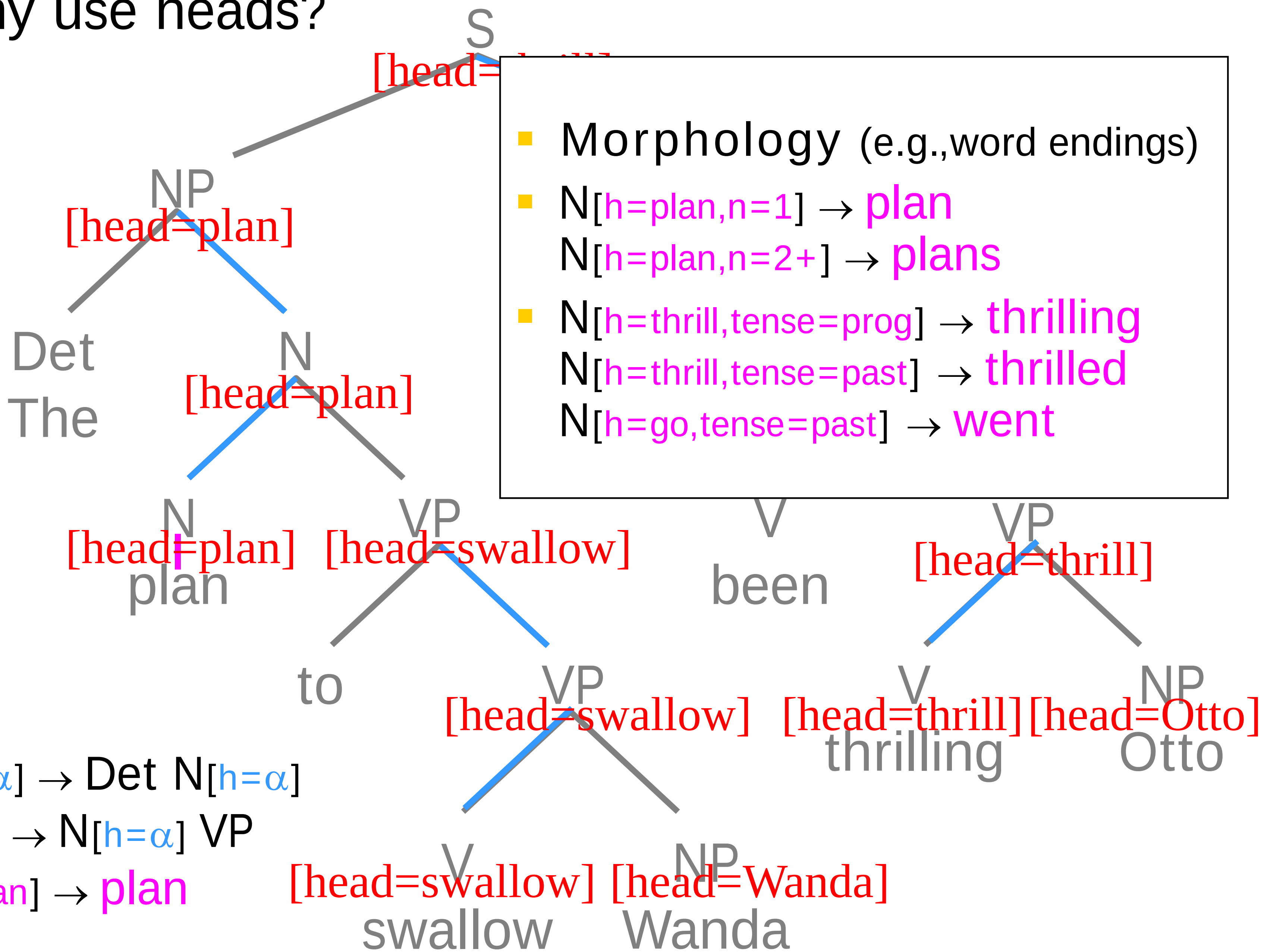


$\mathrm{NP}[h=\alpha] \rightarrow \mathrm{Det}\ \mathrm{N}[h=\alpha]$

$\mathrm{N}[h=\alpha] \rightarrow \mathrm{N}[h=\alpha]\ \mathrm{VP}$

$\mathrm{N}[h=\mathrm{plan}] \rightarrow \mathrm{plan}$

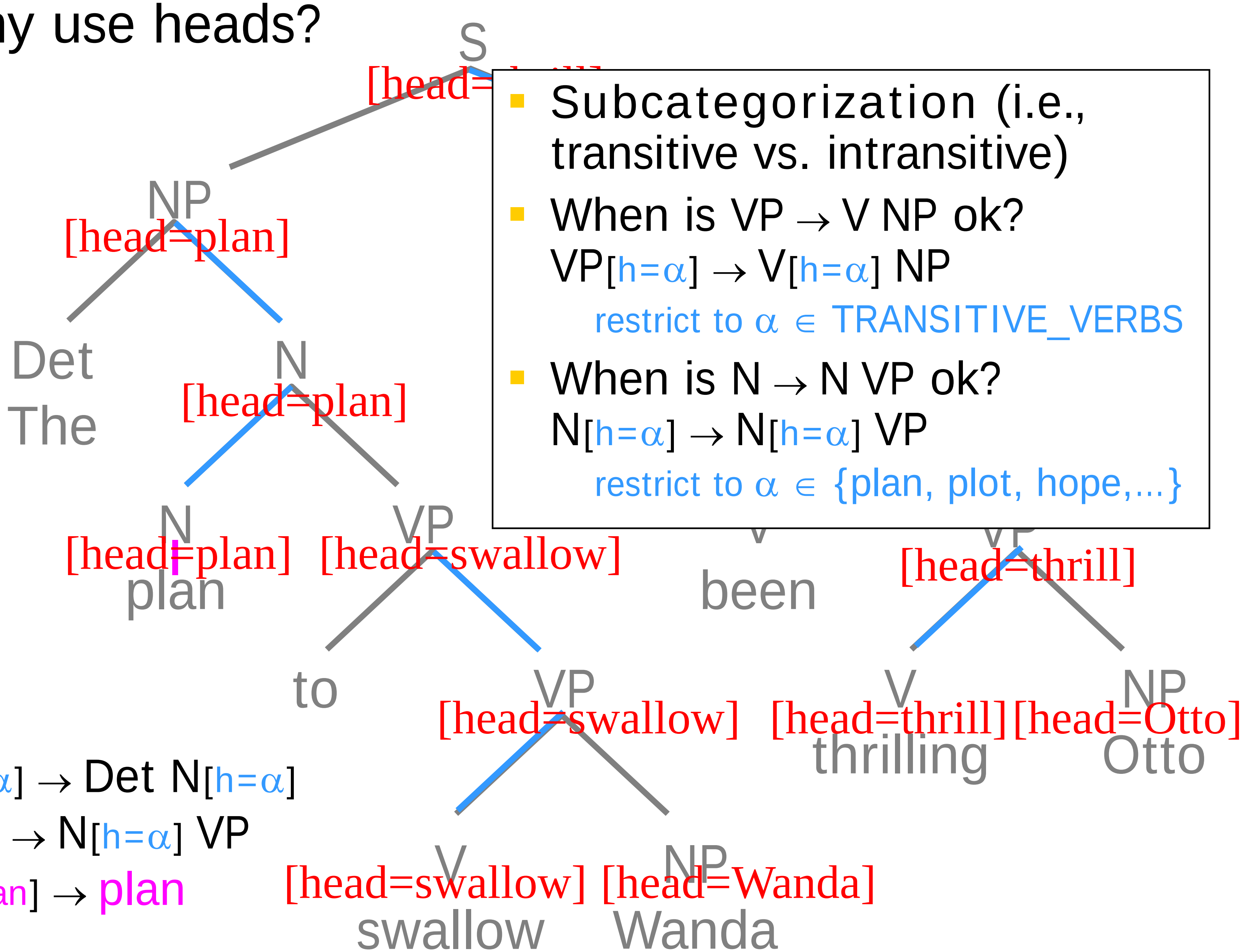
- Why use heads?



- Morphology (e.g.,word endings)
- N[h=plan,n=1] → plan
  N[h=plan,n=2+] → plans
- N[h=thrill,tense=prog] → thrilling
  N[h=thrill,tense=past] → thrilled
  N[h=go,tense=past] → went

NP[h=α] → Det N[h=α]
N[h=α] → N[h=α] VP
N[h=plan] → plan

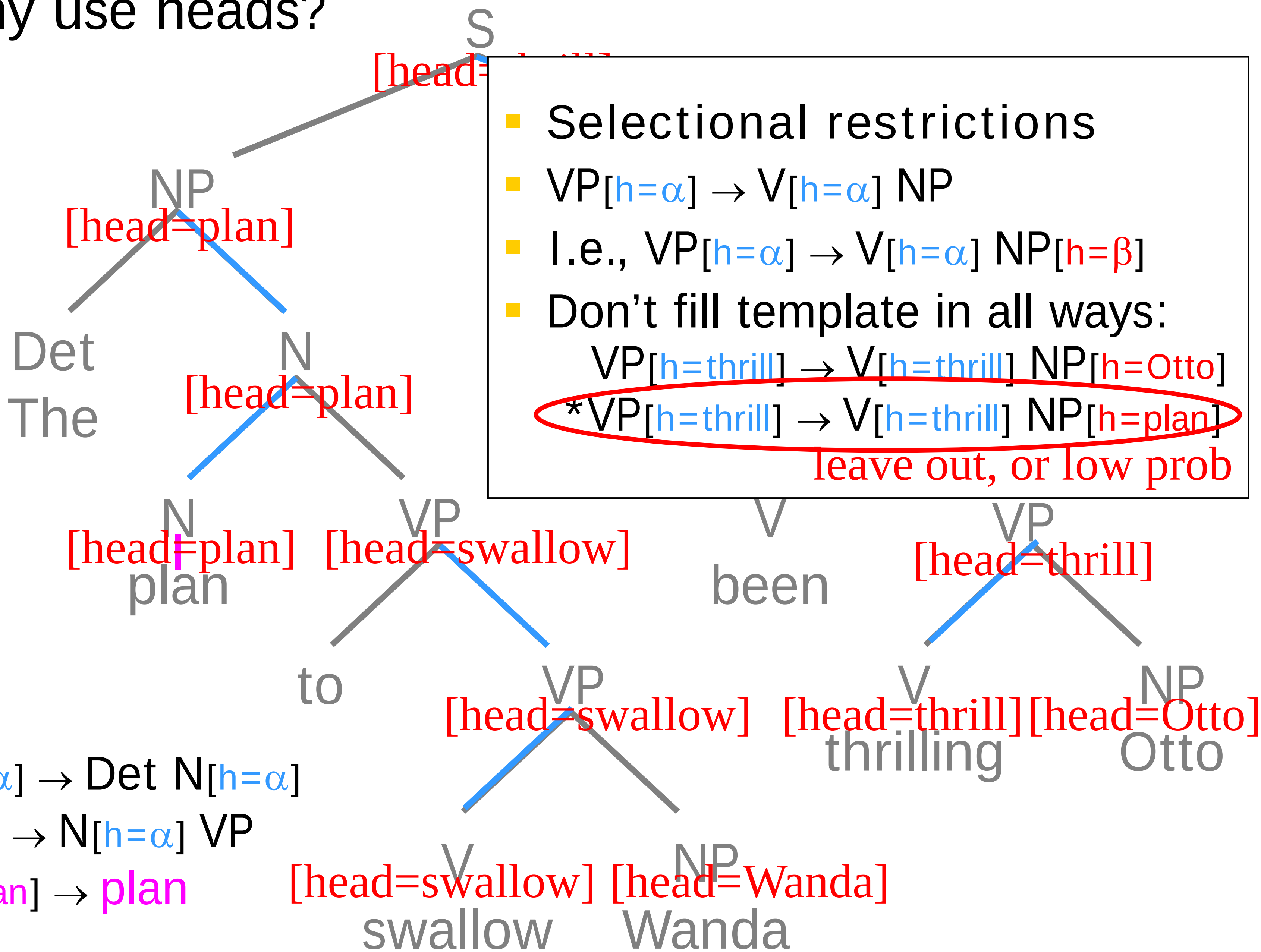
- Why use heads?



- Subcategorization (i.e., transitive vs. intransitive)
- When is VP → V NP ok?
  VP[h=α] → V[h=α] NP
  restrict to α ∈ TRANSITIVE_VERBS
- When is N → N VP ok?
  N[h=α] → N[h=α] VP
  restrict to α ∈ {plan, plot, hope,...}

NP[h=α] → Det N[h=α]
N[h=α] → N[h=α] VP
N[h=plan] → plan

- Why use heads?



- Selectional restrictions
- $\mathrm{VP}[h=\alpha] \rightarrow \mathrm{V}[h=\alpha]\ \mathrm{NP}$
- I.e., $\mathrm{VP}[h=\alpha] \rightarrow \mathrm{V}[h=\alpha]\ \mathrm{NP}[h=\beta]$
- Don't fill template in all ways:
  - $\mathrm{VP}[h=\mathrm{thrill}] \rightarrow \mathrm{V}[h=\mathrm{thrill}]\ \mathrm{NP}[h=\mathrm{Otto}]$
  - $^{*}\mathrm{VP}[h=\mathrm{thrill}] \rightarrow \mathrm{V}[h=\mathrm{thrill}]\ \mathrm{NP}[h=\mathrm{plan}]$
  - leave out, or low prob

$\mathrm{NP}[h=\alpha] \rightarrow \mathrm{Det}\ \mathrm{N}[h=\alpha]$

$\mathrm{N}[h=\alpha] \rightarrow \mathrm{N}[h=\alpha]\ \mathrm{VP}$

$\mathrm{N}[h=\mathrm{plan}] \rightarrow$ plan

Log-Linear Models of Rule Probabilities

- What is the probability of this rule?

  S[head=thrill, tense=pres, ...] $\rightarrow$
  NP[head=plan, num=1, animate=no...]
  VP[head=thrill, tense=pres, num=1, ...]

- We have many related rules.
- p(NP[...] VP[...] | S[...])

  $= (1/Z) \exp \sum_k \theta_k \cdot f_k($S[...] $\rightarrow$ NP[...] VP[...]$)$

- We are choosing among all rules that expand S[...].
- If a rule has positively-weighted features, they raise its probability. Negatively-weighted features lower it.
- Which features fire will depend on the attributes!

Log-Linear Models of Rule Probabilities

S[head=thrill, tense=pres, ...] $\rightarrow$
NP[head=plan, num=1, animate=no, ...]
VP[head=thrill, tense=pres, num=1, ...]

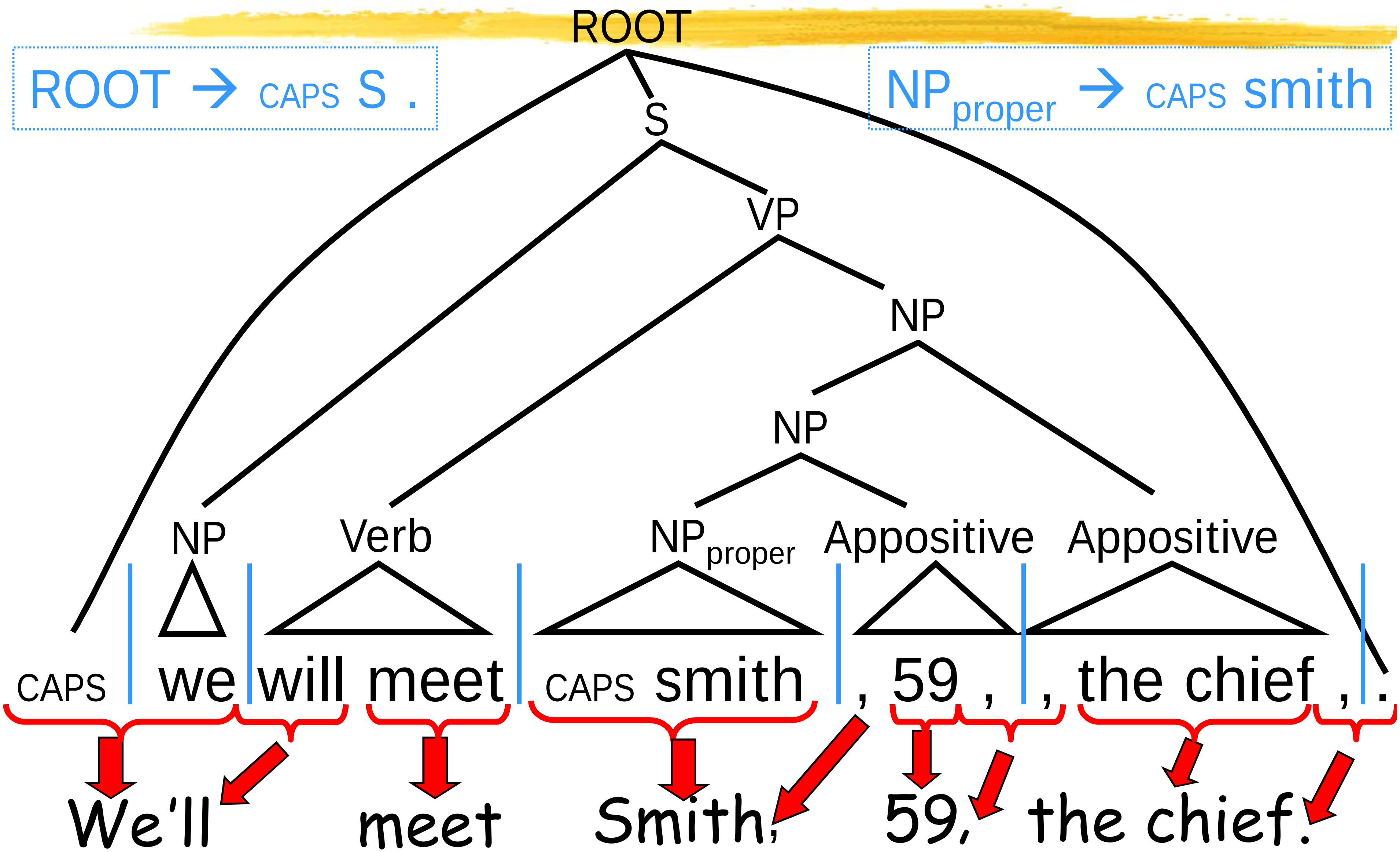
- Some features that might fire on this ...
  - The raw rule without attributes is S $\rightarrow$ NP VP.
    - *Is that good? Does this feature have positive weight?*
  - The NP and the VP agree in number. *Is that good?*
  - The head of the NP is "plan." *Is that good?*
  - The verb "thrill" will get a subject.
  - The verb "thrill" will get an inanimate subject.
  - The verb "thrill" will get a subject headed by "plan."
    - *Is that good? Is "plan" a good subject for "thrill"?*

Post-Processing

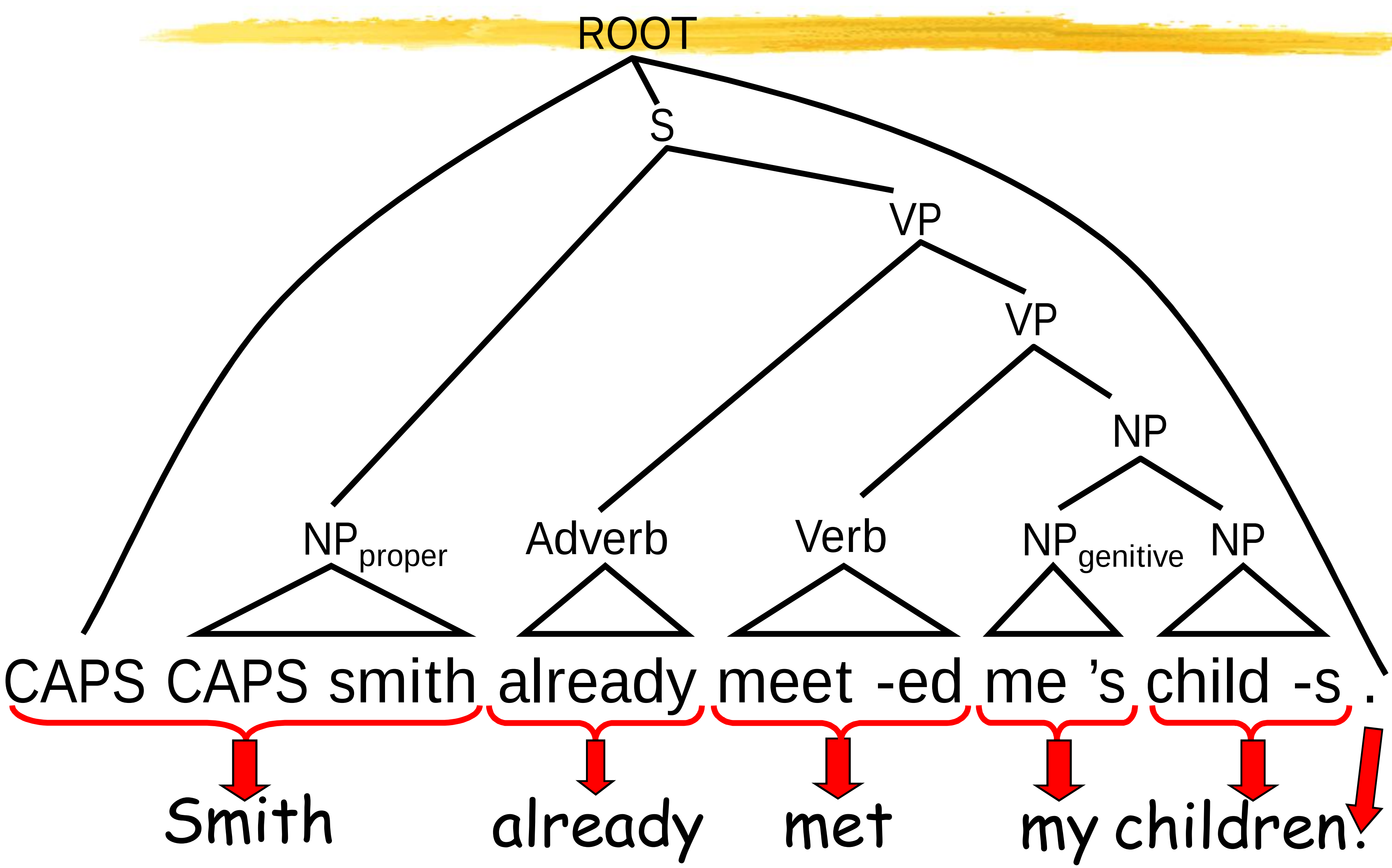
- You don't have to handle everything with tons of attributes on the nonterminals
- Sometimes easier to compose your grammar with a post-processor:
  1. Use your CFG + randsent to generate some convenient internal version of the sentence.
  2. Run that sentence through a post-processor to clean it up for external presentation.
  3. The post-processor can even fix stuff up across constituent boundaries!

We'll see a good family of postprocessors later: finite-state transducers.

Simpler Grammar + Post-Processing



Simpler Grammar + Post-Processing



What Do These Enhancements Give You? And What Do They Cost?

- In a sense, nothing and nothing!
  - Can automatically convert our new fancy CFG to an old plain CFG.
- This is reassuring ...
  - We haven't gone off into cloud-cuckoo land where "ooh, look what languages I can invent."
    - Even fancy CFGs can't describe crazy non-human languages such as the language consisting only of prime numbers.
    - Because we already know that plain CFGs can't do that.
  - We can still use our old algorithms, randsent and parse.
    - Just convert to a plain CFG and run the algorithms on that.
- But we do get a benefit!
  - Attributes and post-processing allow simpler grammars.
  - Same log-linear features are shared across many rules.
  - A language learner thus has fewer things to learn.

Analogy: What Does Dyna Give You?

- In a sense, nothing and nothing!
  - We can automatically convert our fancy Dyna program to plain old machine code.
- This is reassuring ...
  - A standard computer can still run Dyna.
    No special hardware or magic wands are required.
- But we do get a benefit!
  - High-level programming languages allow shorter programs that are easier to write, understand, and modify.

What Do These Enhancements Give You? And What Do They Cost?

- In a sense, nothing and nothing!
  - We can automatically convert our new fancy CFG to an old plain CFG.
- Nonterminals with attributes ➔ more nonterminals
  - S[head=α, tense=β] $\rightarrow$ NP[num=γ,...] VP[head=α, tense=β, num=γ...]
  - Can write out versions of this rule for all values of α, β, γ
  - Now rename NP[num=1,...] to NP_num_1_...
  - So we just get a plain CFG with a ton of rules and nonterminals
- Post-processing ➔ more nonterminal attributes
  - Example: Post-processor changes "a" to "an" before a vowel
  - But we could handle this using a "starts with vowel" attribute instead
    - The determiner must "agree" with the vowel status of its Nbar
  - This kind of conversion can always be done! (automatically!)
    - At least for post-processors that are finite-state transducers
  - And then we can convert these attributes to nonterminals as above

Part of the English Tense System

	Present	Past	Future	Infinitive
Simple	eats	ate	will eat	to eat
Perfect	has eaten	had eaten	will have eaten	to have eaten
progressive	is eating	was eating	will be eating	to be eating
Perfect+ progressive	has been eating	had been eating	will have been eating	to have been eating

Tenses by Post-Processing: "Affix-hopping" (Chomsky)

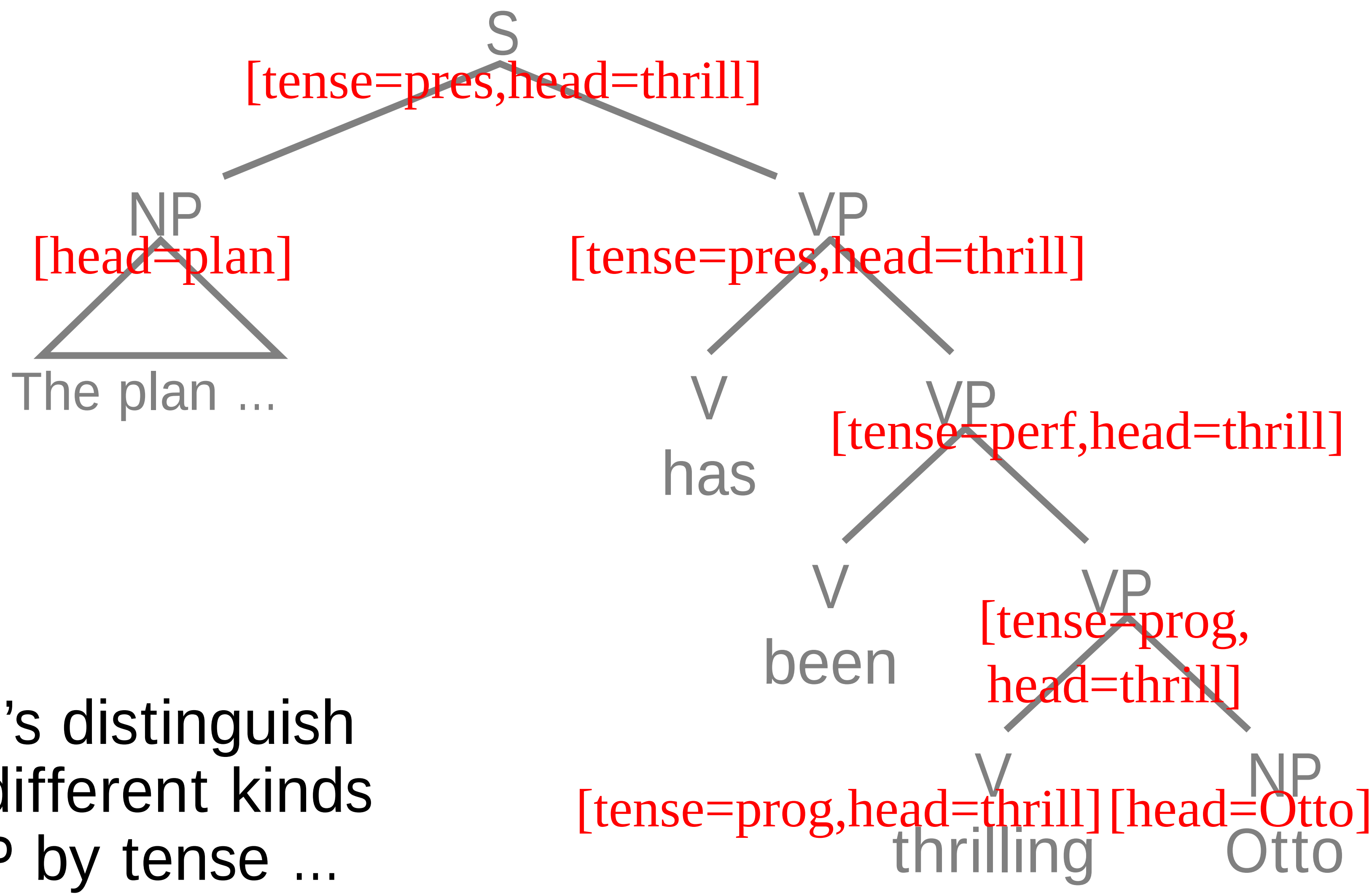
Mary jumps	Mary [–s jump]
Mary has jumped	Mary [-s have] [-en jump]
Mary is jumping	Mary [-s be] [-ing jump]
Mary has been jumping	Mary [-s have] [-en be] [-ing jump]

Agreement, meaning

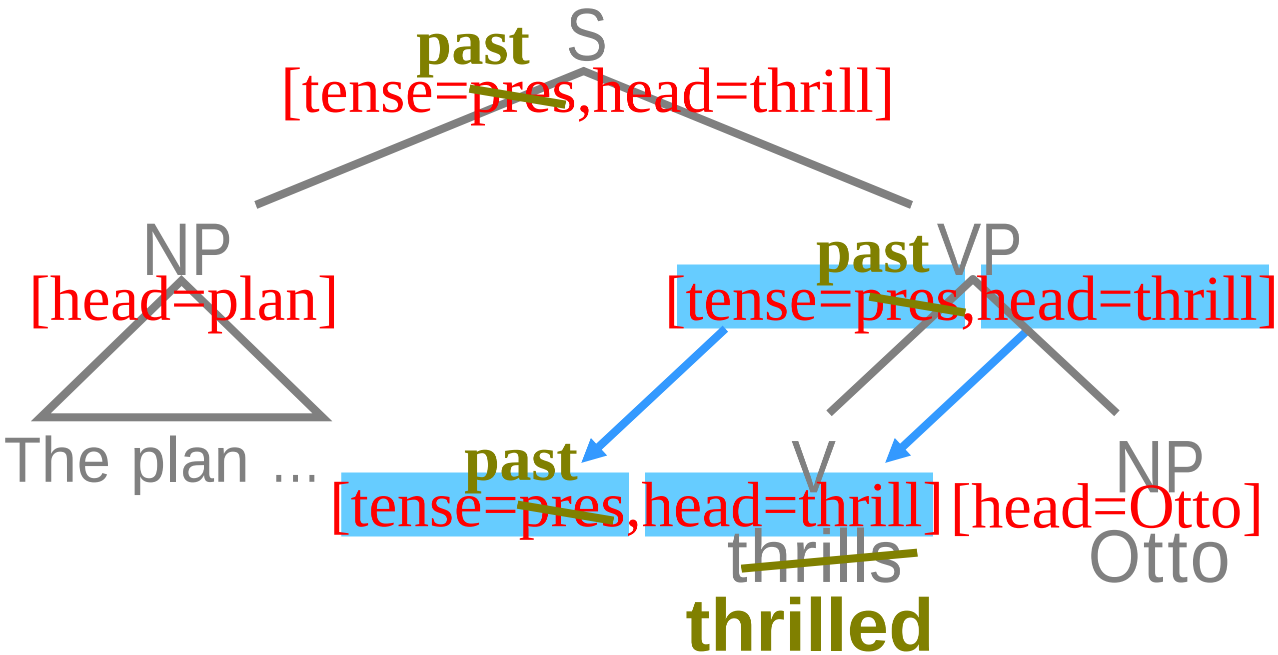
where

- -s denotes "3rd person singular present tense" on following verb (by an –s suffix)
- -en denotes "past participle" (often uses –en or –ed suffix)
- -ing denotes "present participle"   Etc.

Could we instead describe the patterns via attributes?

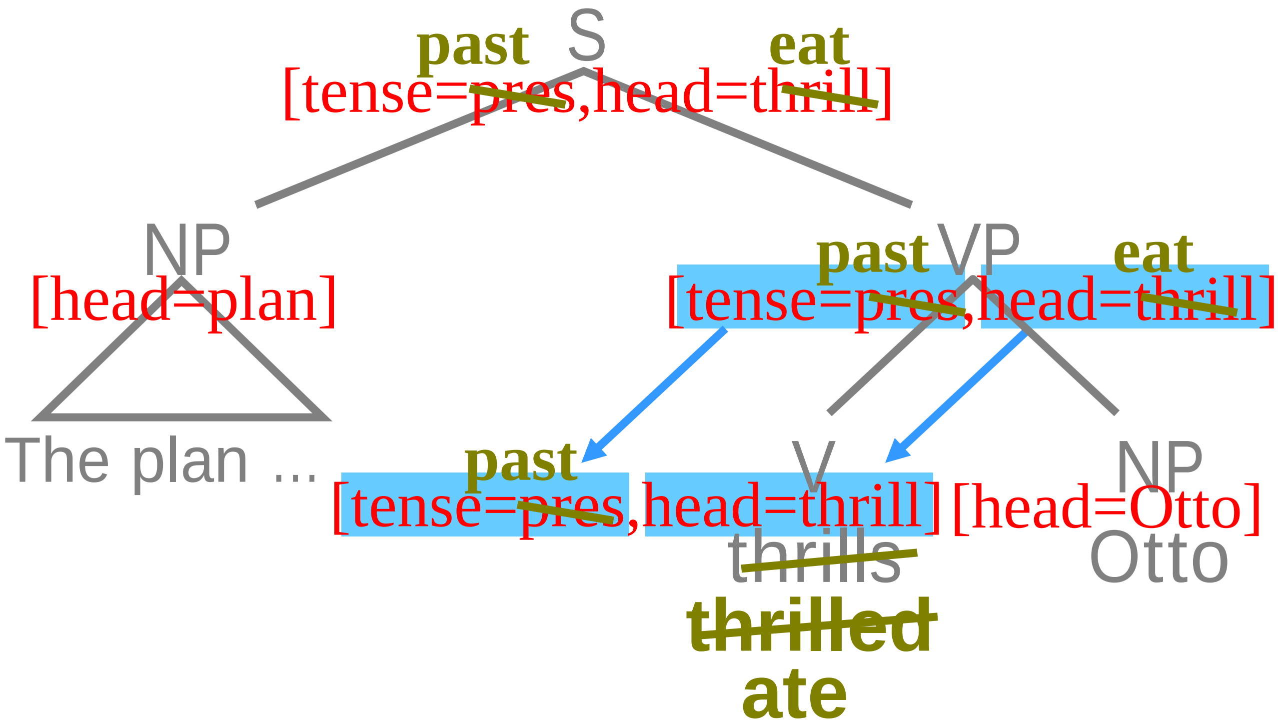


- Let’s distinguish the different kinds of VP by tense ...

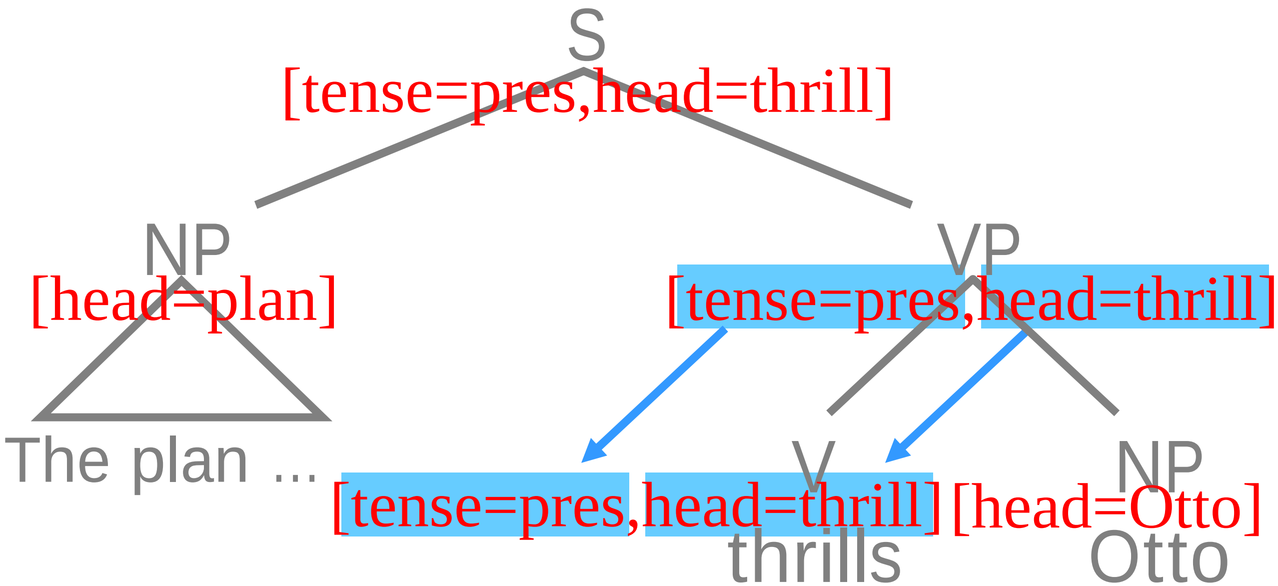


Past

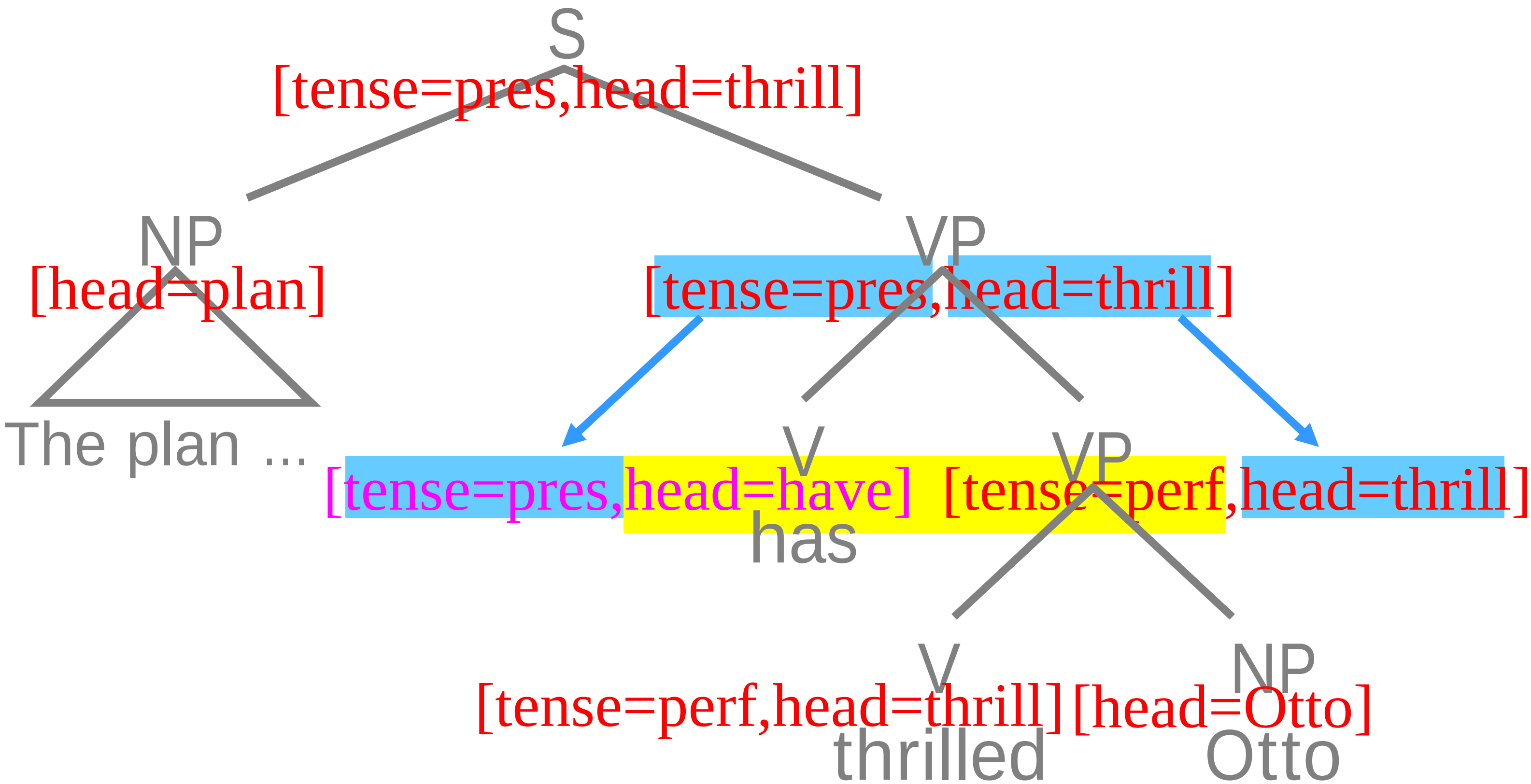
- ~~Present~~ tense



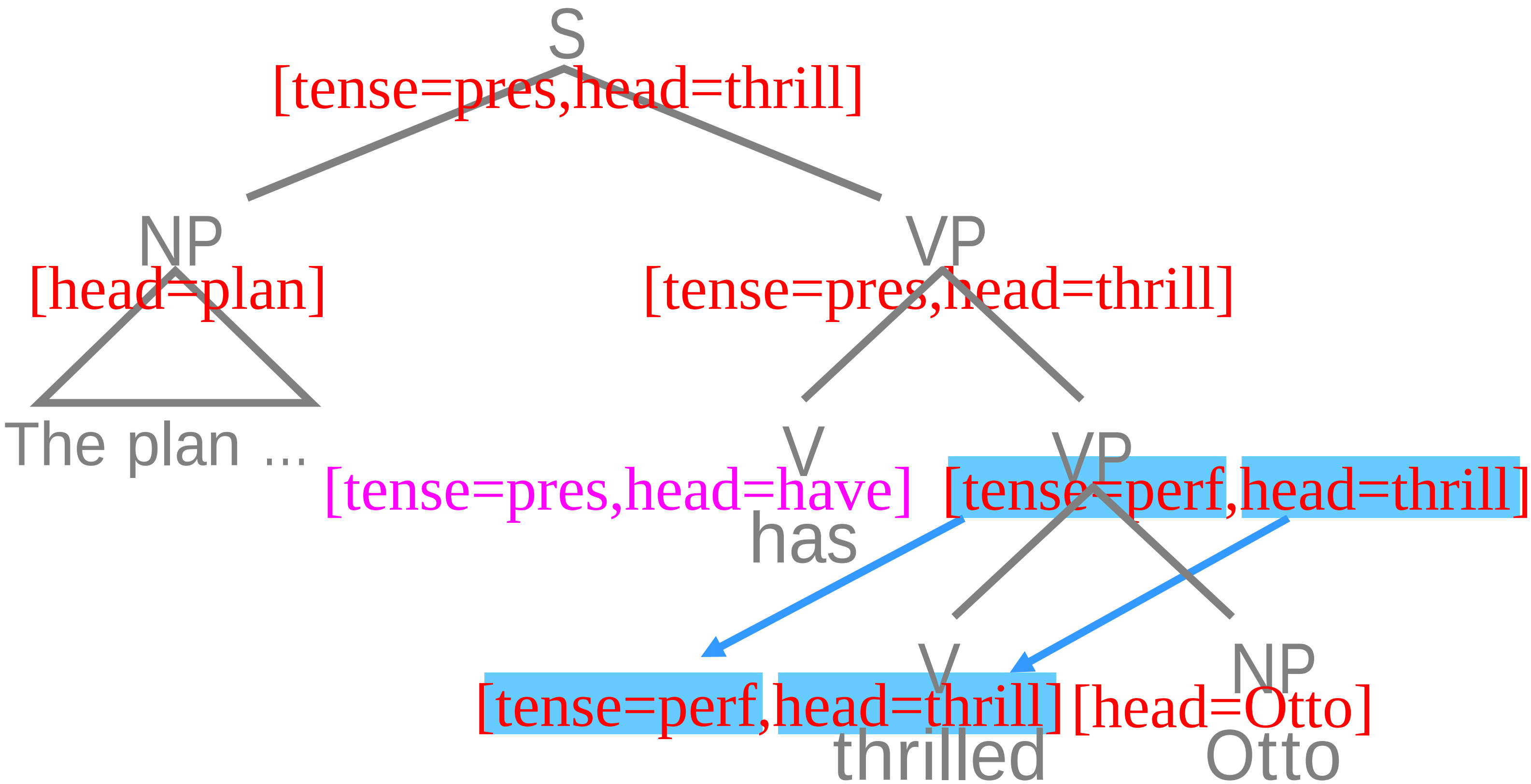
Past
- ~~Present~~ tense



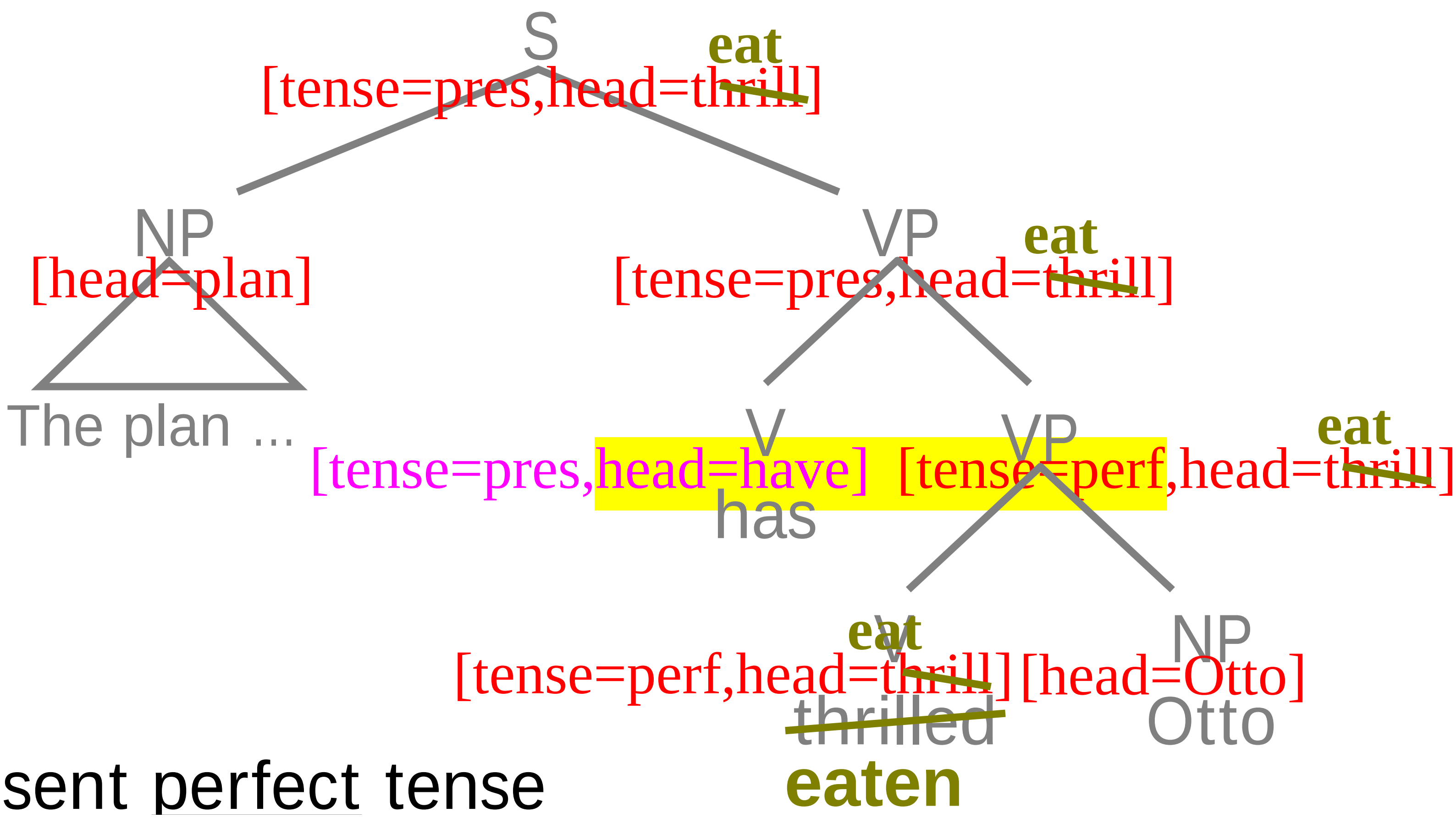
- Present tense (again)



- Present <u>perfect</u> tense

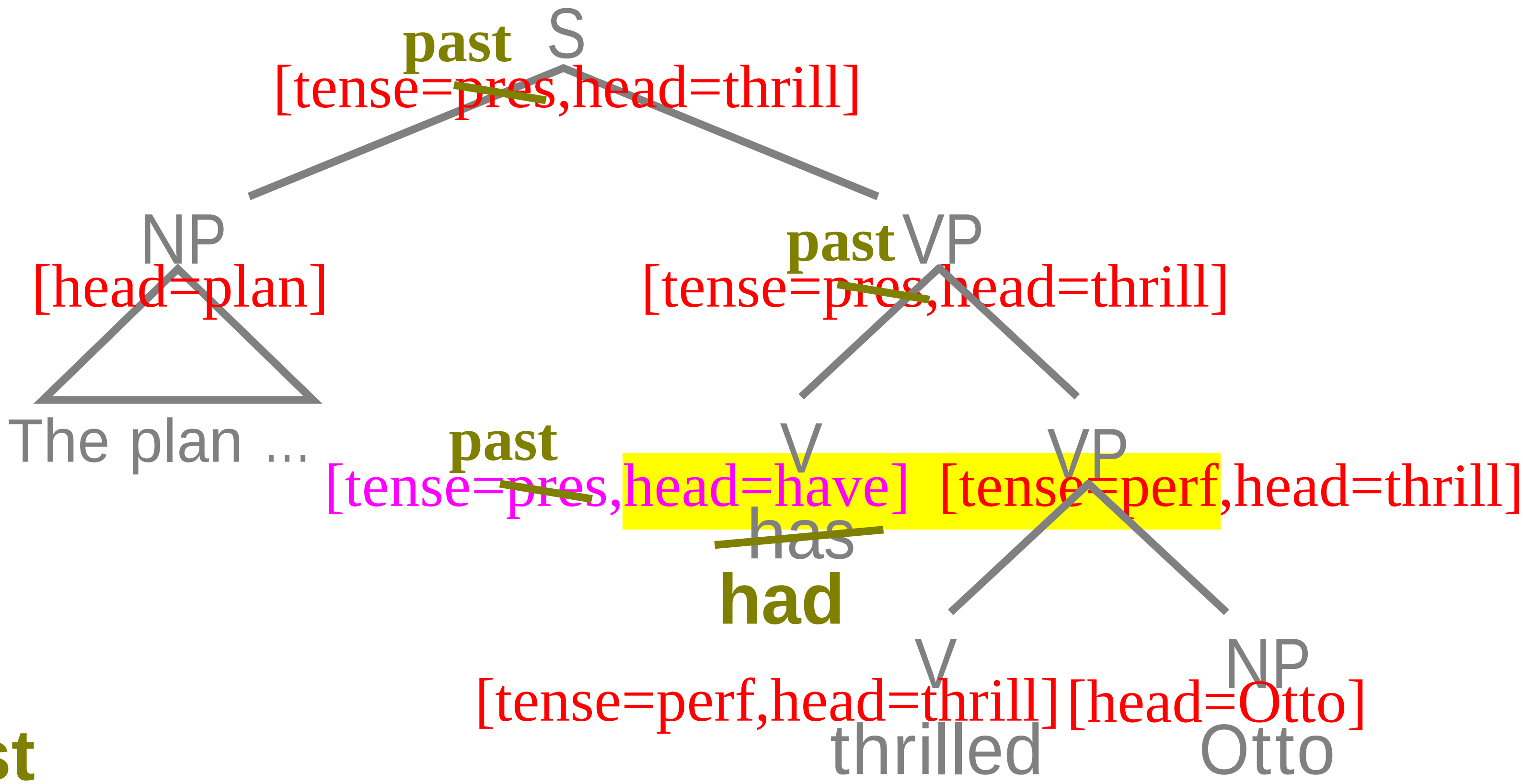


- Present perfect tense



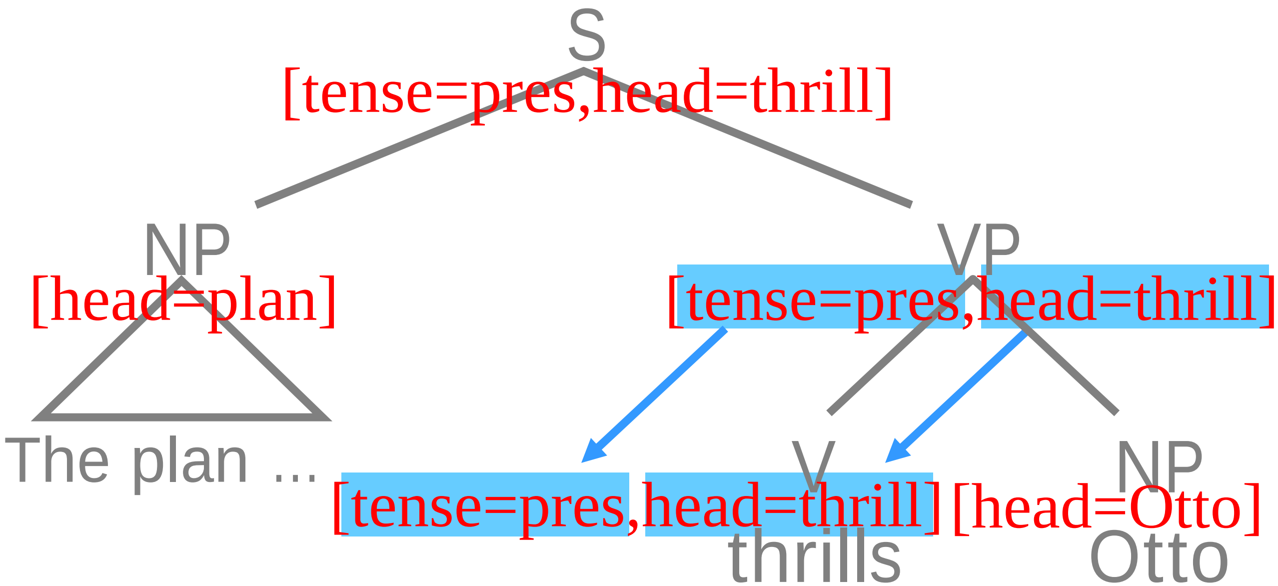
- Present perfect tense
- The yellow material makes it a perfect tense – what effects?

not **ate** – why?

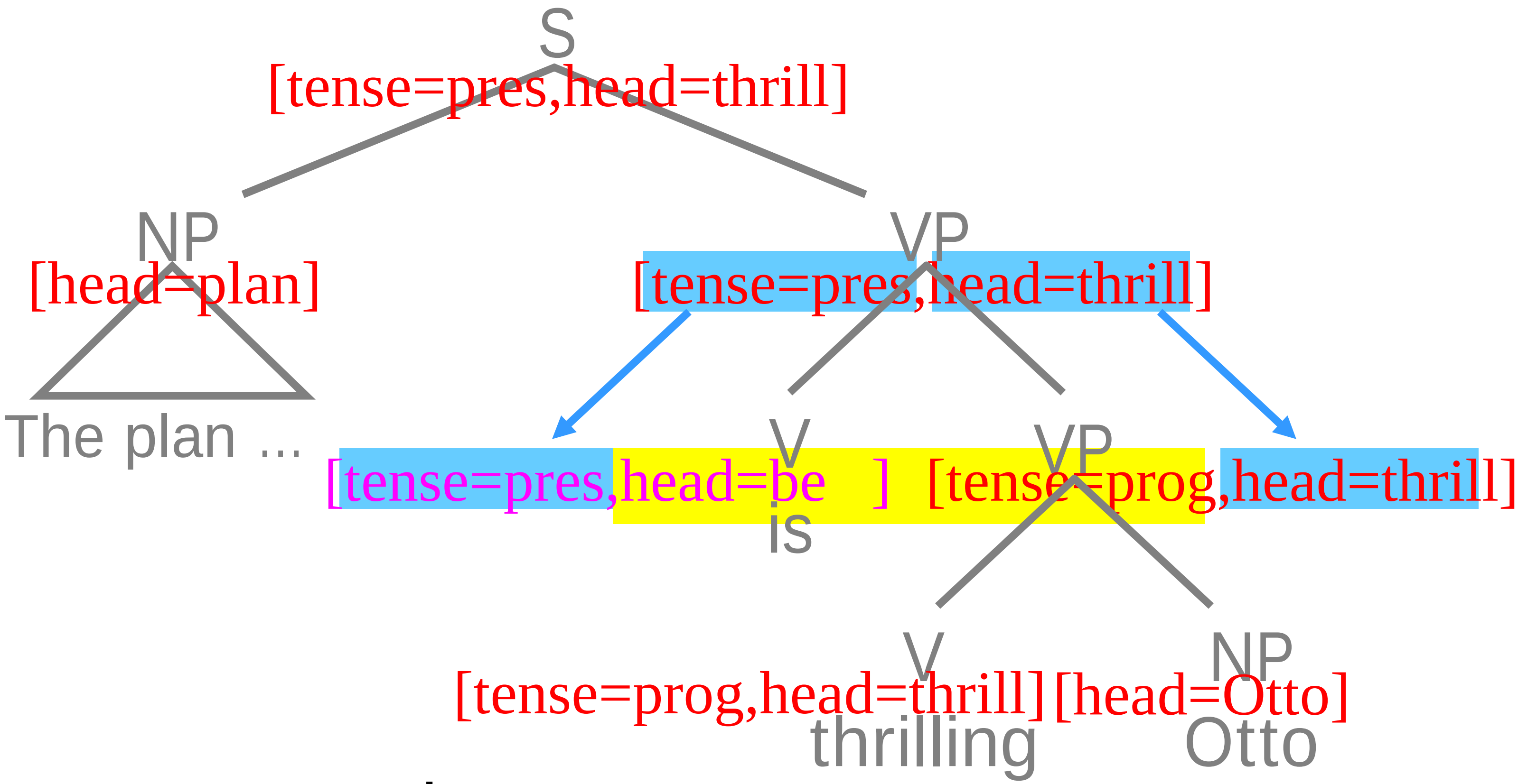


Past

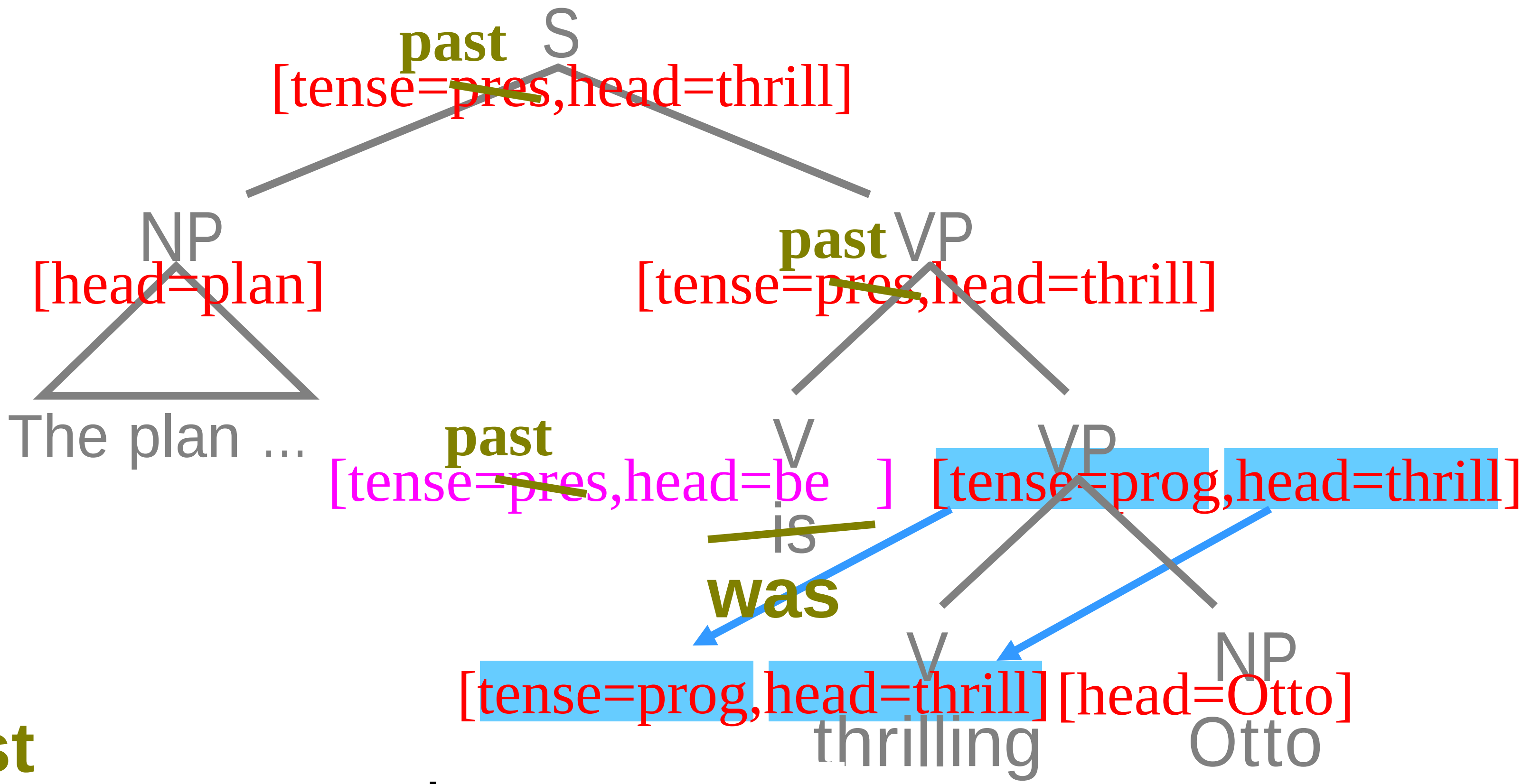
- ~~Present~~ perfect tense



- Present tense (again)

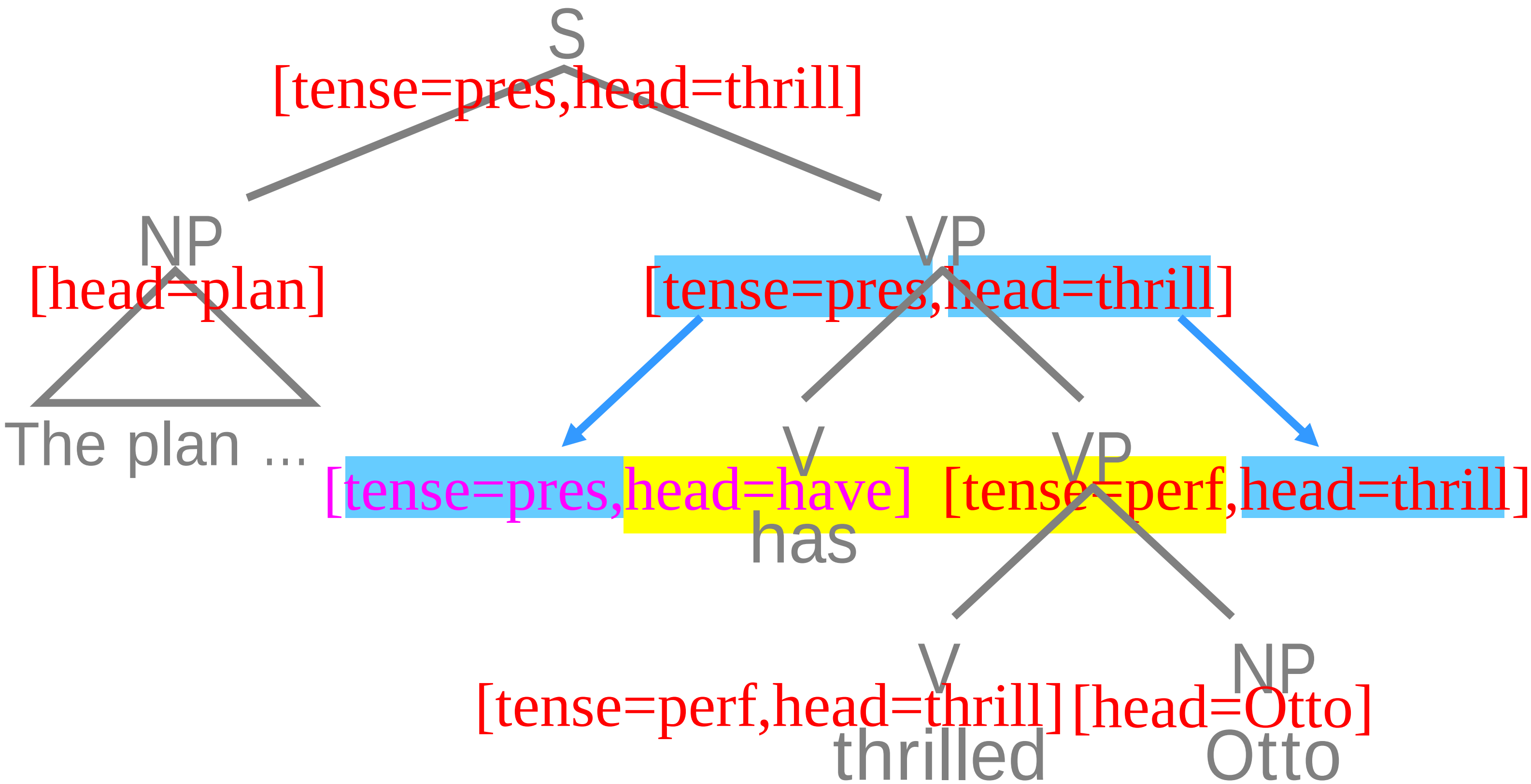


- Present <u>progressive</u> tense

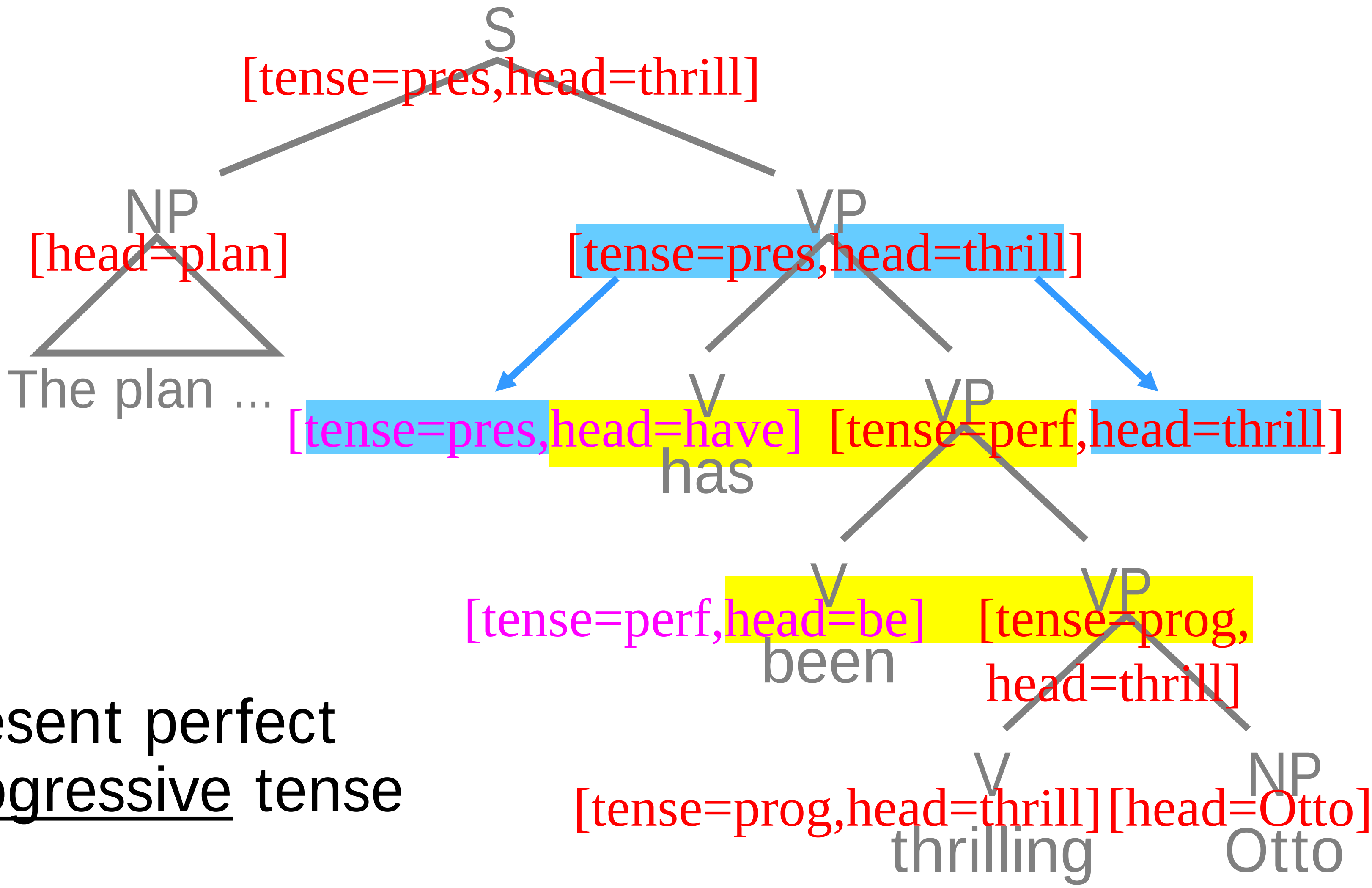


Past

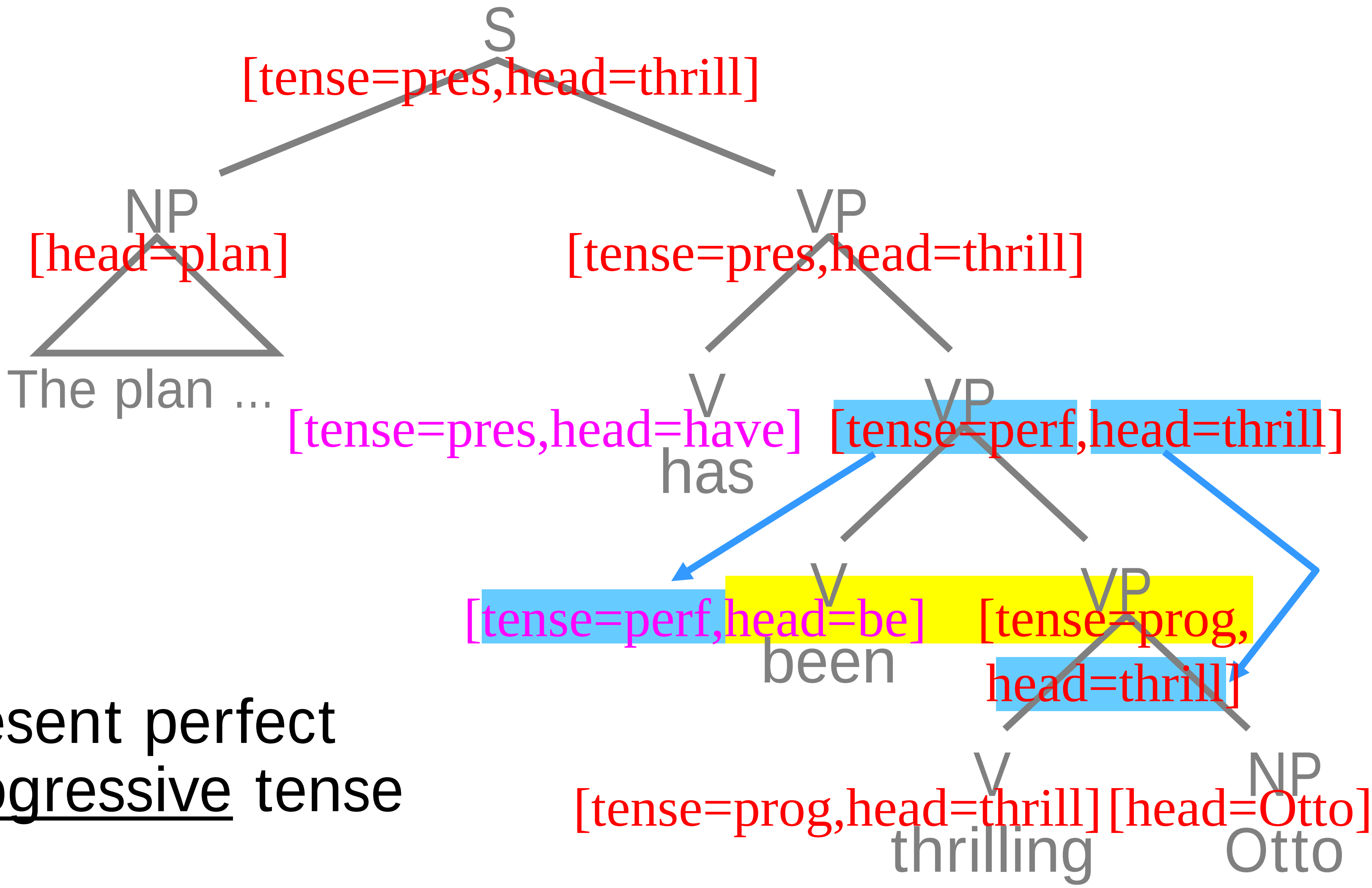
- ~~Present~~ progressive tense



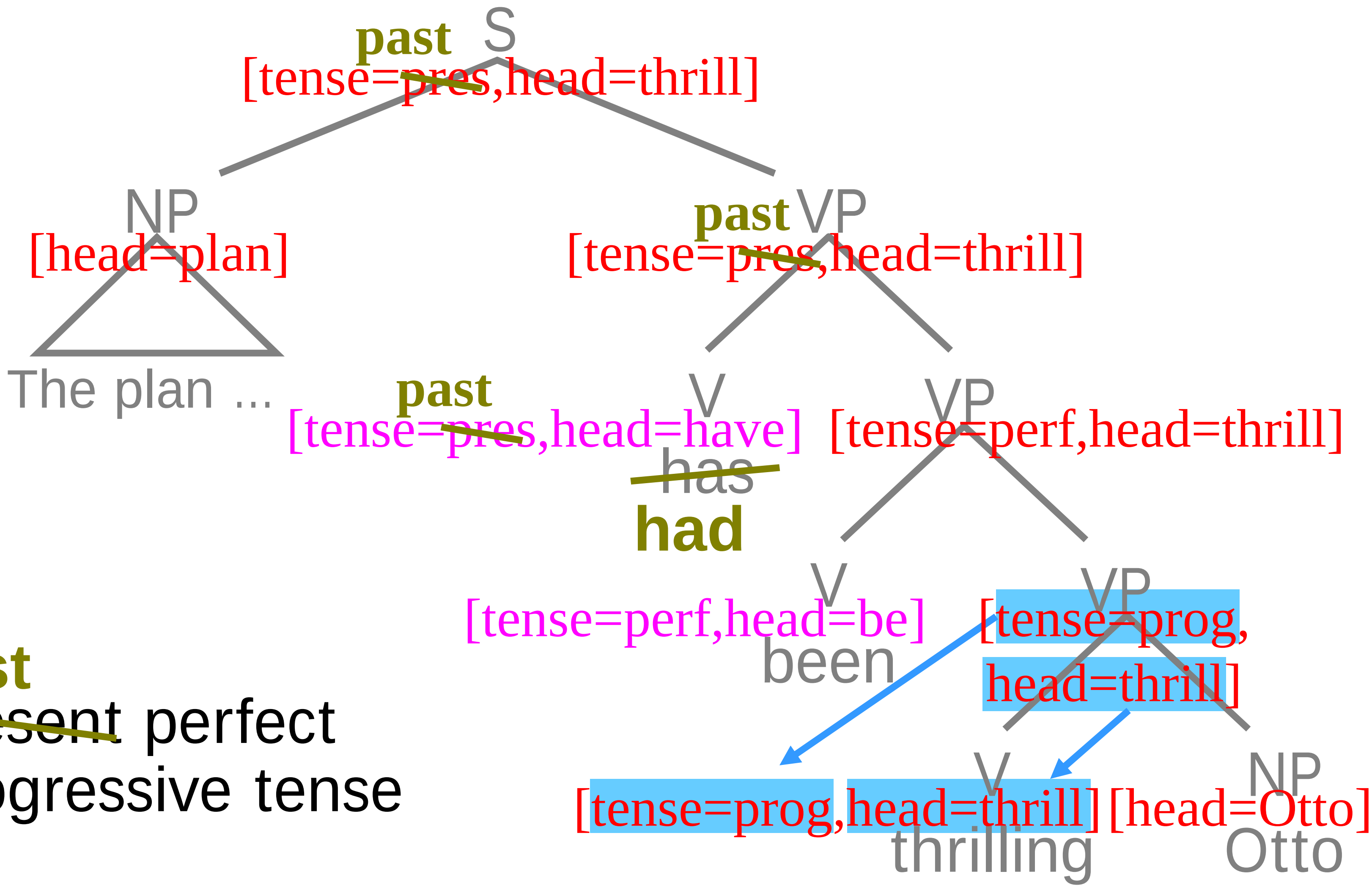
- Present perfect tense (again)



- Present perfect <u>progressive</u> tense

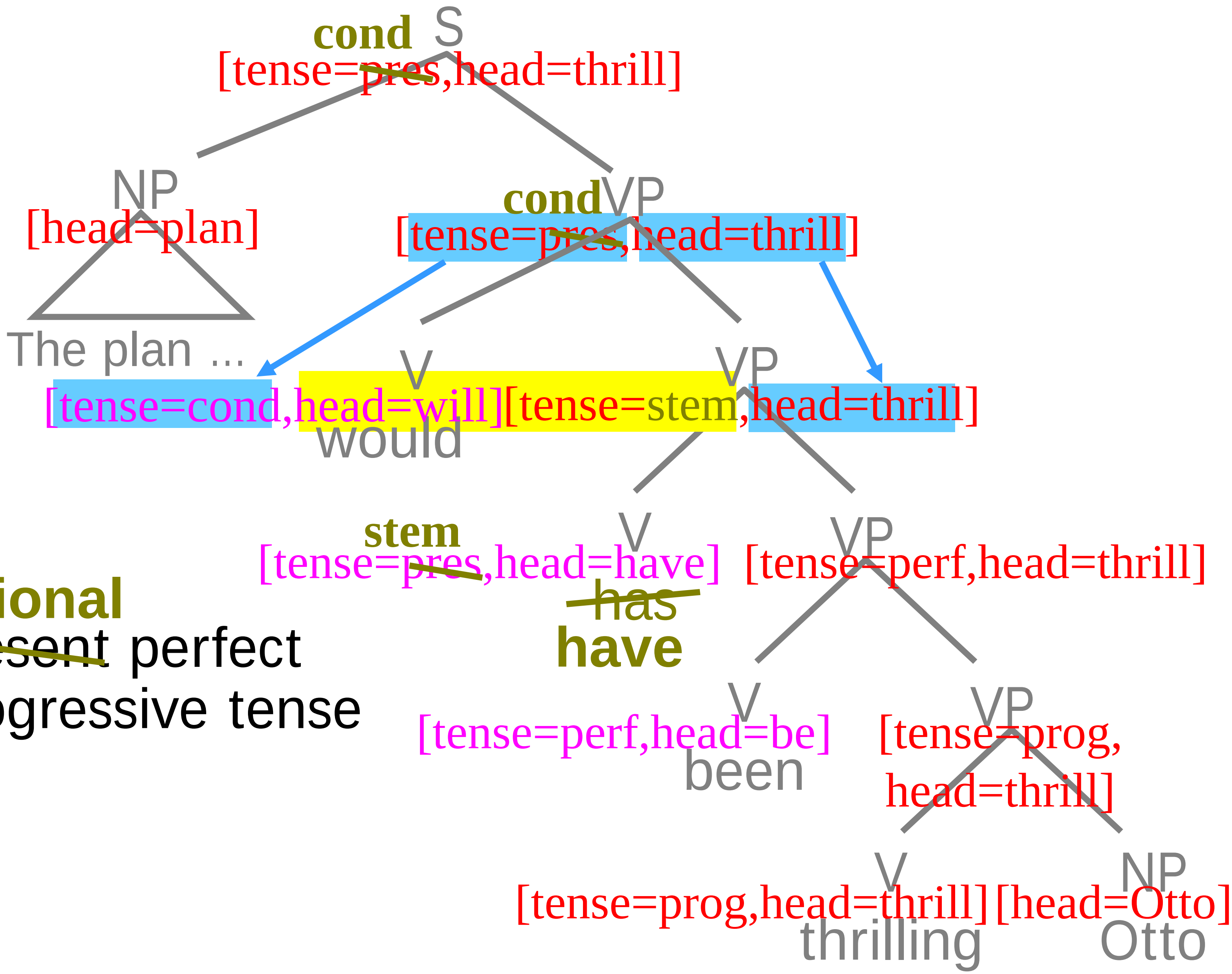


- Present perfect <u>progressive</u> tense



Past

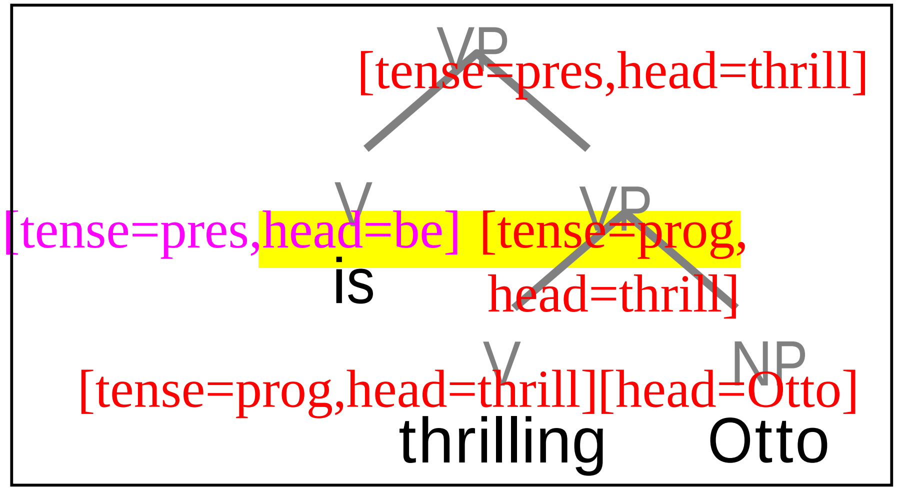
- ~~Present~~ perfect progressive tense

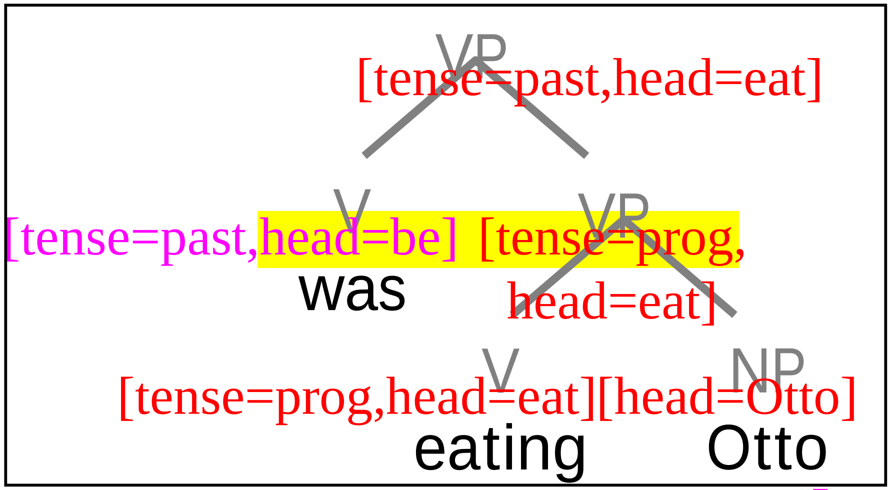


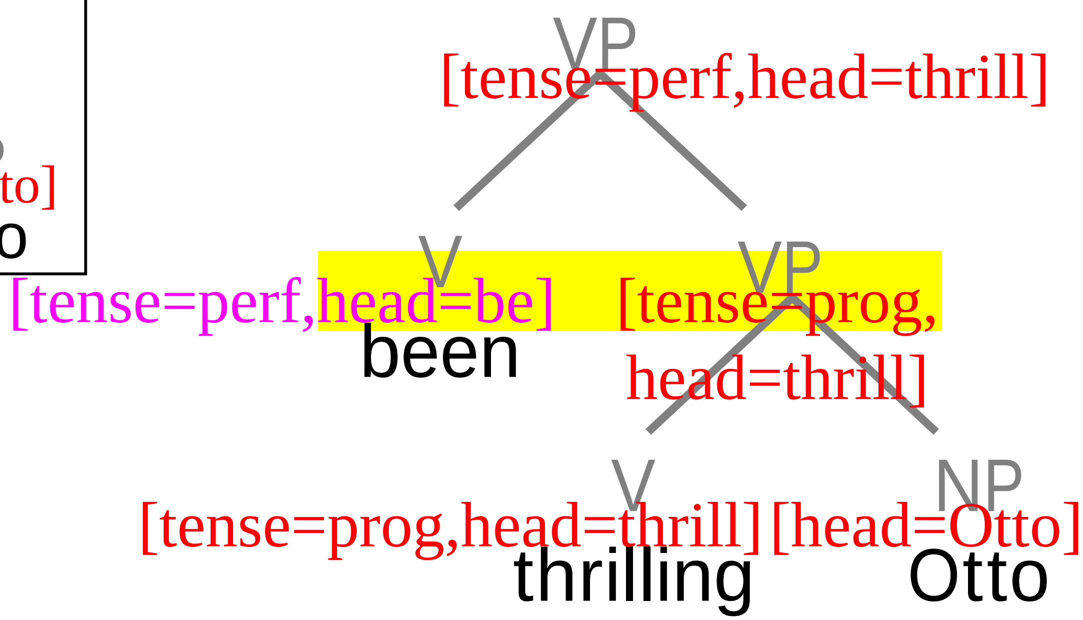
Conditional

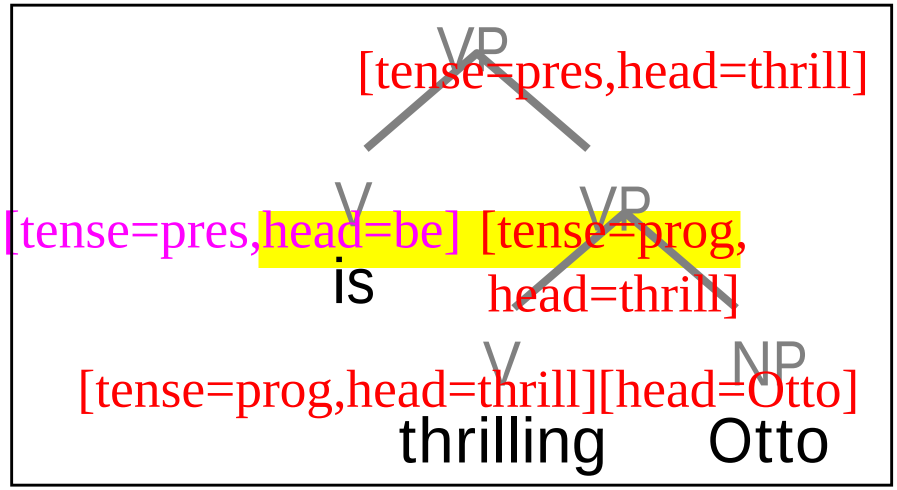
- ~~Present~~ perfect progressive tense

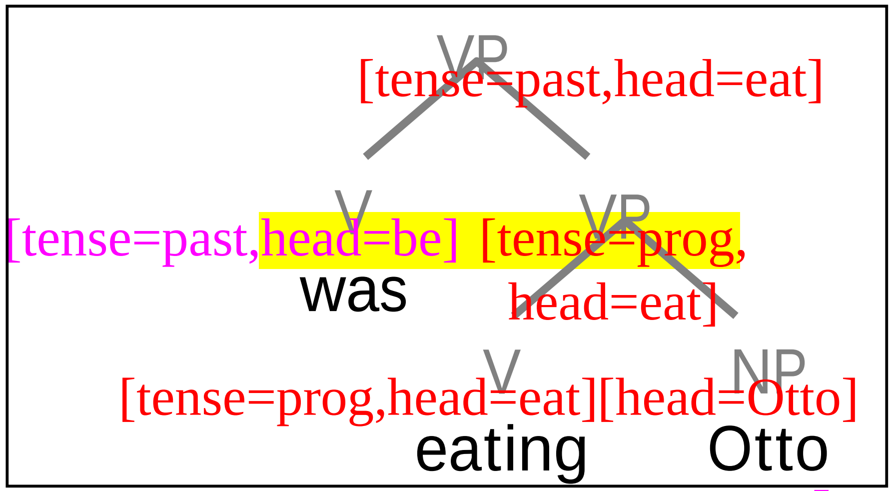
- So what pattern do all progressives follow?



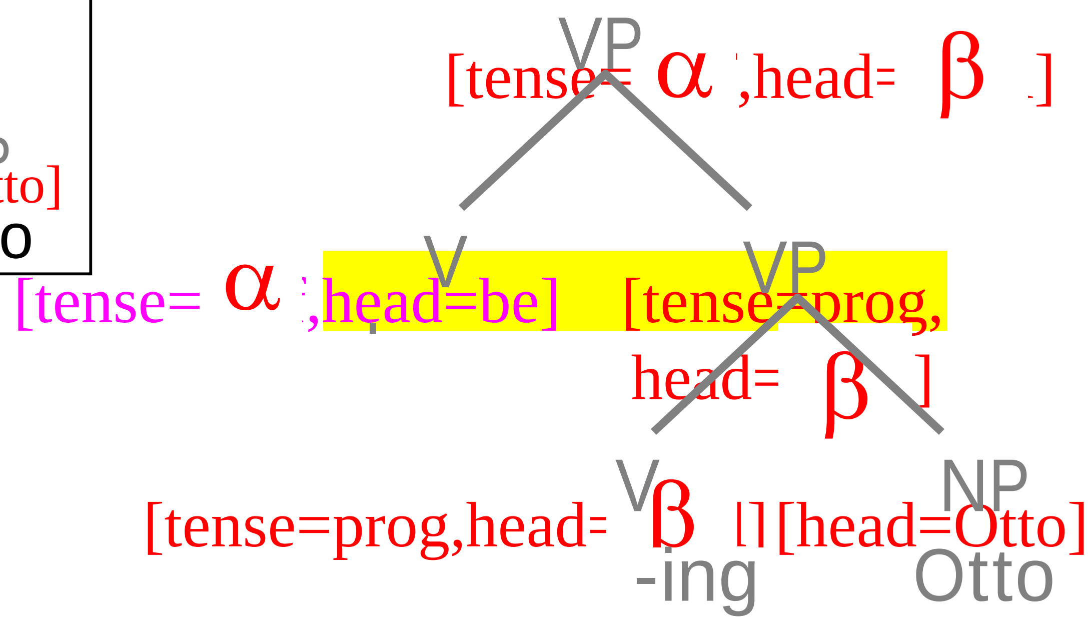








- So what pattern do all progressives follow?



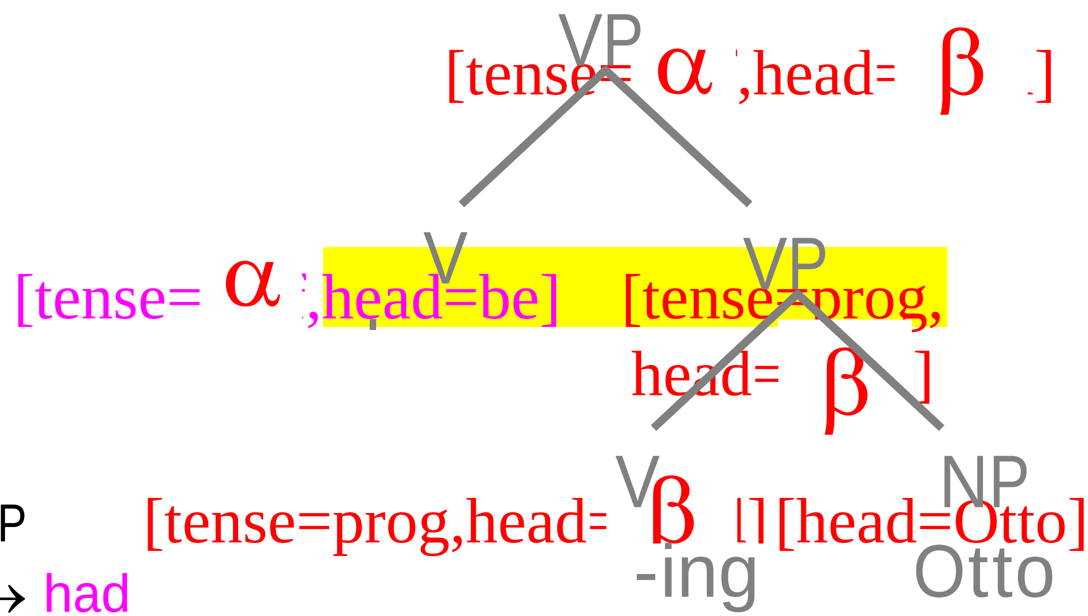
Progressive: VP[tense=α, head=β, ...] $\rightarrow$ V[tense=α, head=be ...] VP[tense=prog, head=β ...]

Perfect: VP[tense=α, head=β, ...] $\rightarrow$ V[tense=α, head=have ...] VP[tense=perf, head=β ...]

Future or conditional: VP[tense=α, head=β, ...] $\rightarrow$ V[tense=α, head=will ...] VP[tense=stem, head=β ...]

Infinitive: VP[tense=inf, head=β, ...] $\rightarrow$ to VP[tense=stem, head=β ...]

Etc.



As well as the “ordinary” rules:

VP[tense=α, head=β, ...]
$\rightarrow$ V[tense=α, head=β, ...] NP

V[tense=past, head=have ...] $\rightarrow$ had

Gaps ("deep" grammar!)

- Pretend "kiss" is a pure transitive verb.
- Is "the president kissed" grammatical?
  - If so, what type of phrase is it?

- the sandwich that
- I wonder what
- What else has

the president kissed e
Sally said the president kissed e
Sally consumed the pickle with e
Sally consumed e with the pickle

Gaps

- Object gaps:
- the sandwich that
- I wonder what
- What else has

the president kissed e
Sally said the president kissed e
Sally consumed the pickle with e
Sally consumed e with the pickle

[how could you tell the difference?]

- Subject gaps:
- the sandwich that
- I wonder what
- What else has

e kissed the president
Sally said e kissed the president

Gaps

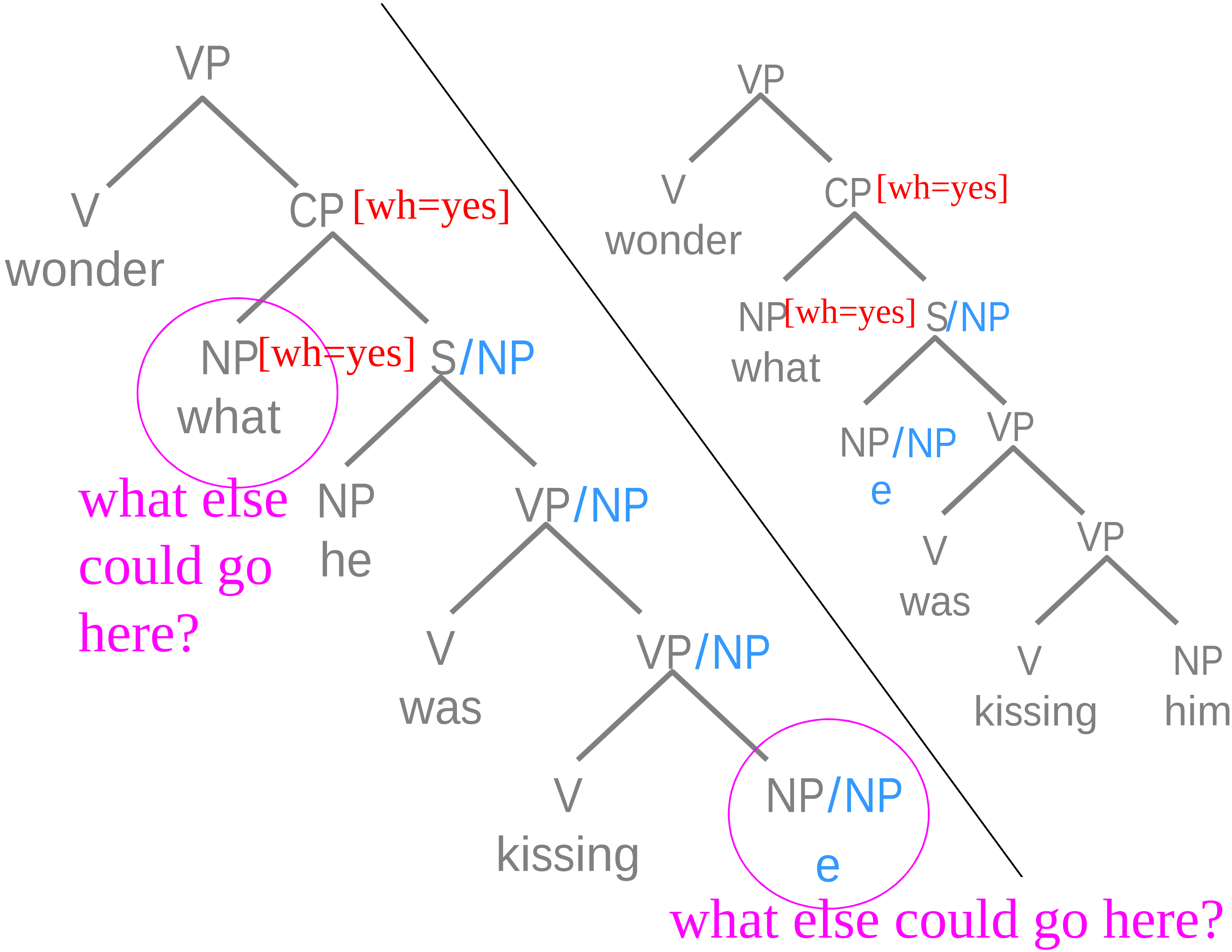
- All gaps are really the same – a missing NP:
- the sandwich that
- I wonder what
- What else has

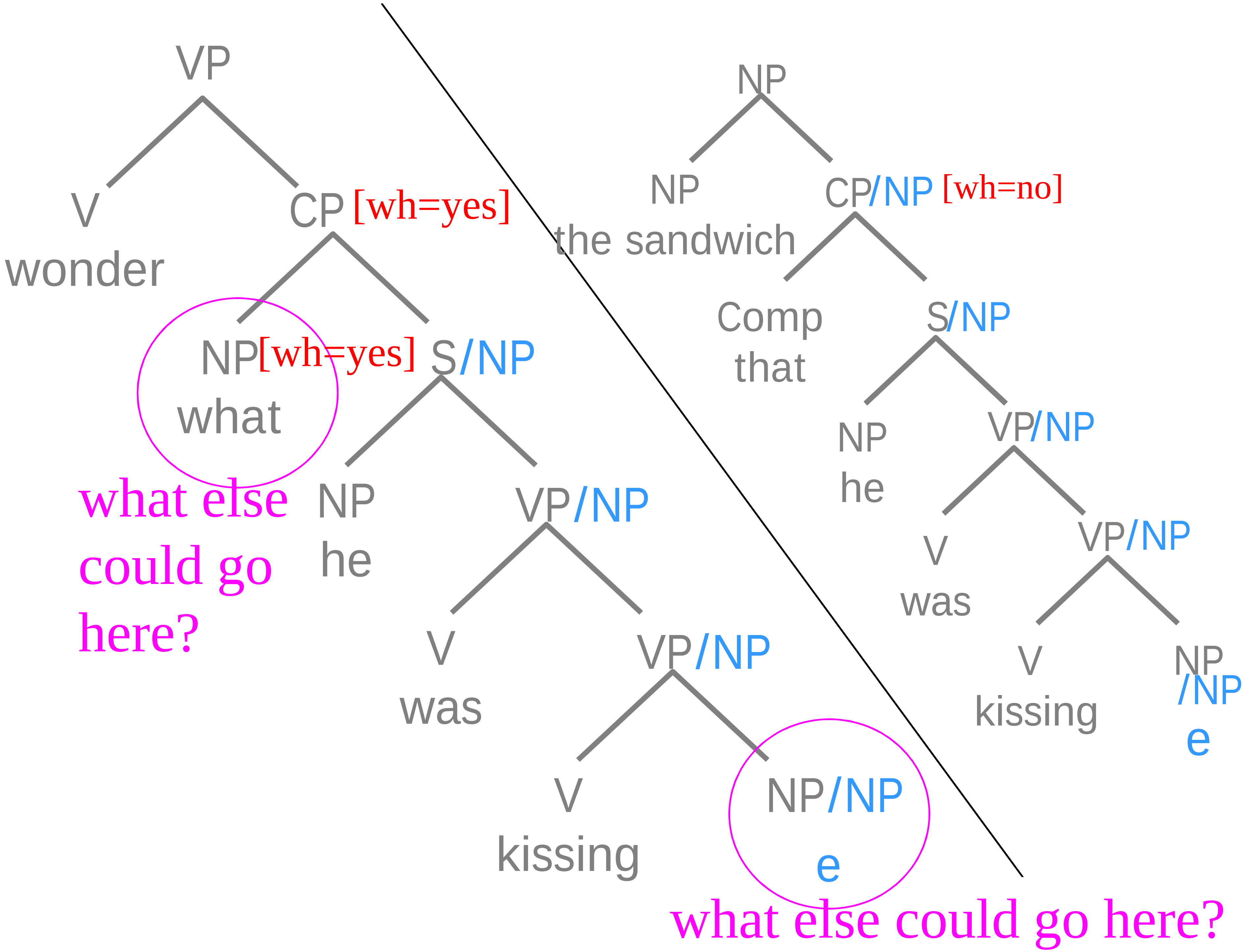
the president kissed e
Sally said the president kissed e
Sally consumed the pickle with e
Sally consumed e with the pickle
e kissed the president
Sally said e kissed the president

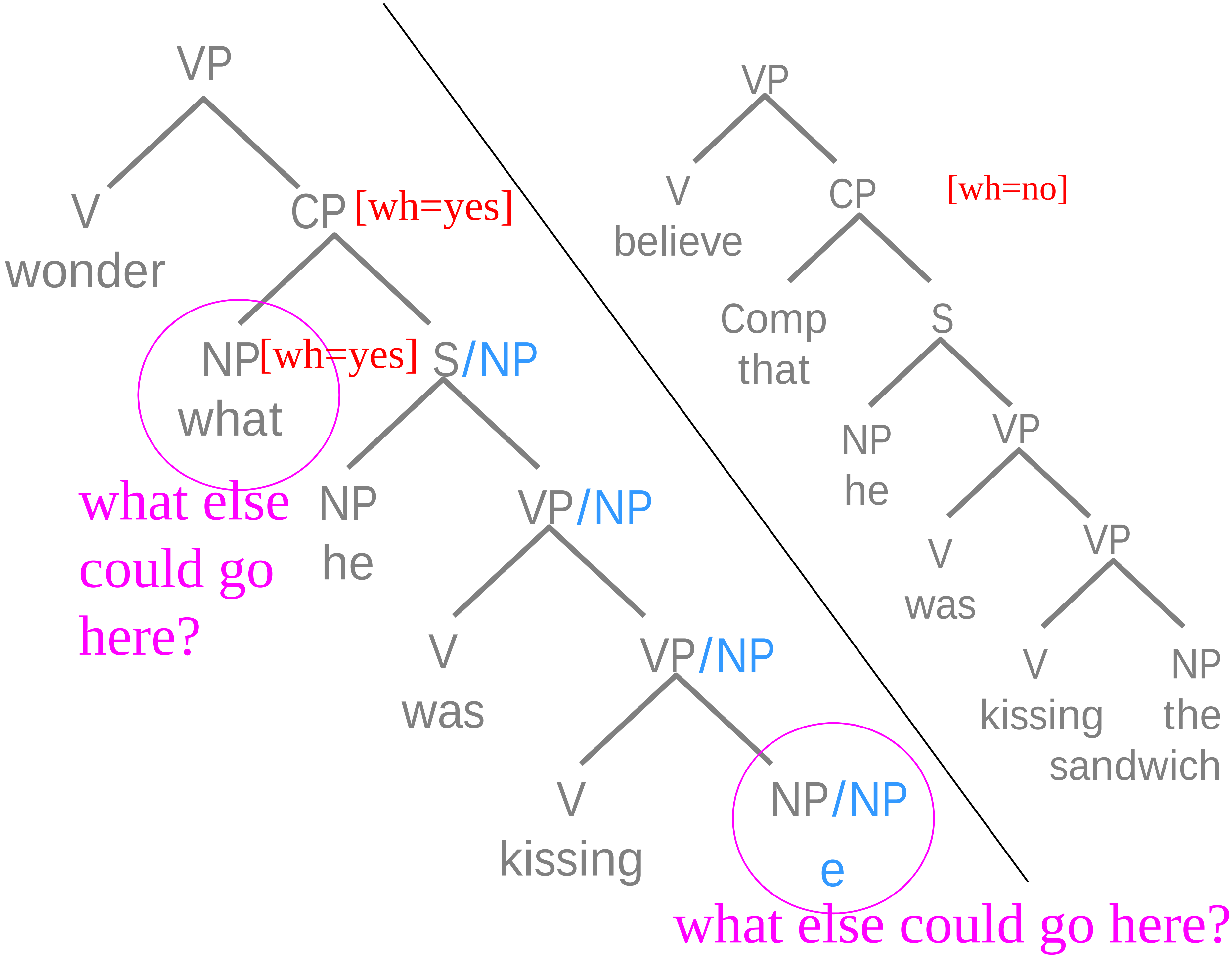
Phrases with missing NP:

X[missing=NP]

or just X/NP for short

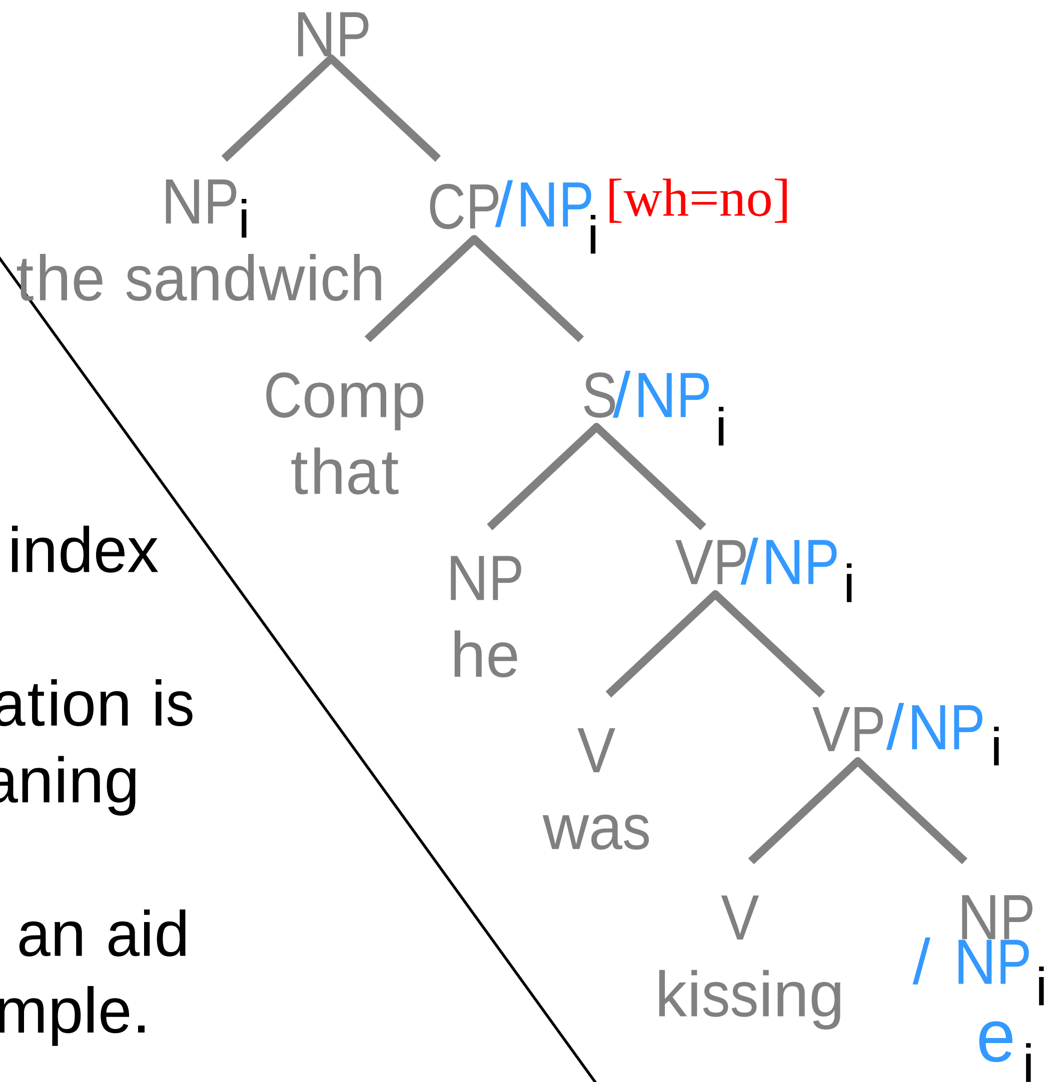

VP
V
wonder
CP [wh=yes]
NP[wh=yes]
what
S/NP
what else could go here?
NP
he
VP/NP
V
was
VP/NP
V
kissing
NP/NP
e
what else could go here?
VP
V
wonder
CP [wh=yes]
NP[wh=yes]
what
S/NP
NP/NP
e
VP
V
was
VP
V
kissing
NP
him


VP
V
wonder
CP [wh=yes]
NP[wh=yes]
what
S/NP
what else could go here?
NP
he
VP/NP
V
was
VP/NP
V
kissing
NP/NP
e
what else could go here?
NP
NP
the sandwich
CP/NP [wh=no]
Comp
that
S/NP
NP
he
VP/NP
V
was
VP/NP
V
kissing
NP
/NP
e



VP
V
wonder
CP [wh=yes]
NP[wh=yes]
what
S/NP
what else could go here?
NP
he
VP/NP
V
was
VP/NP
V
kissing
NP/NP
e
what else could go here?
VP
V
believe
CP
[wh=no]
Comp
that
S
NP
he
VP
V
was
VP
V
kissing
NP
the sandwich

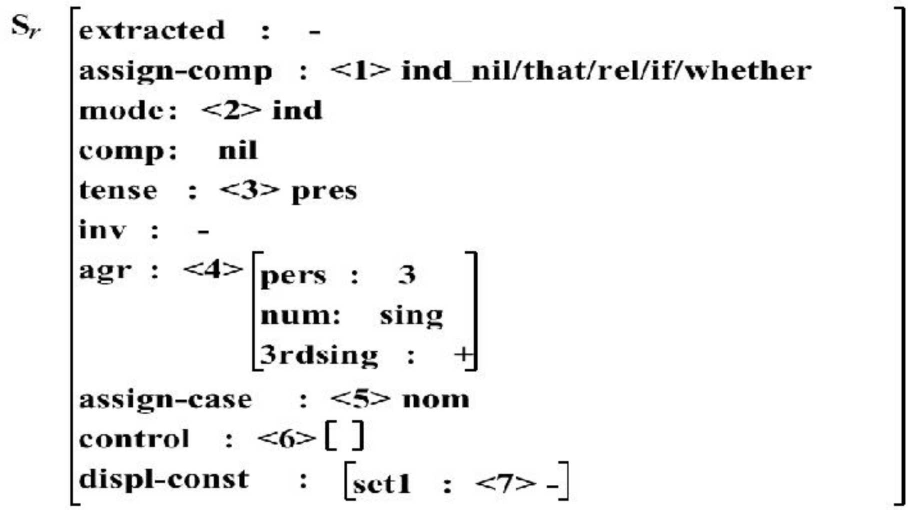
To indicate what fills a gap, people sometimes "coindex" the gap and its filler.

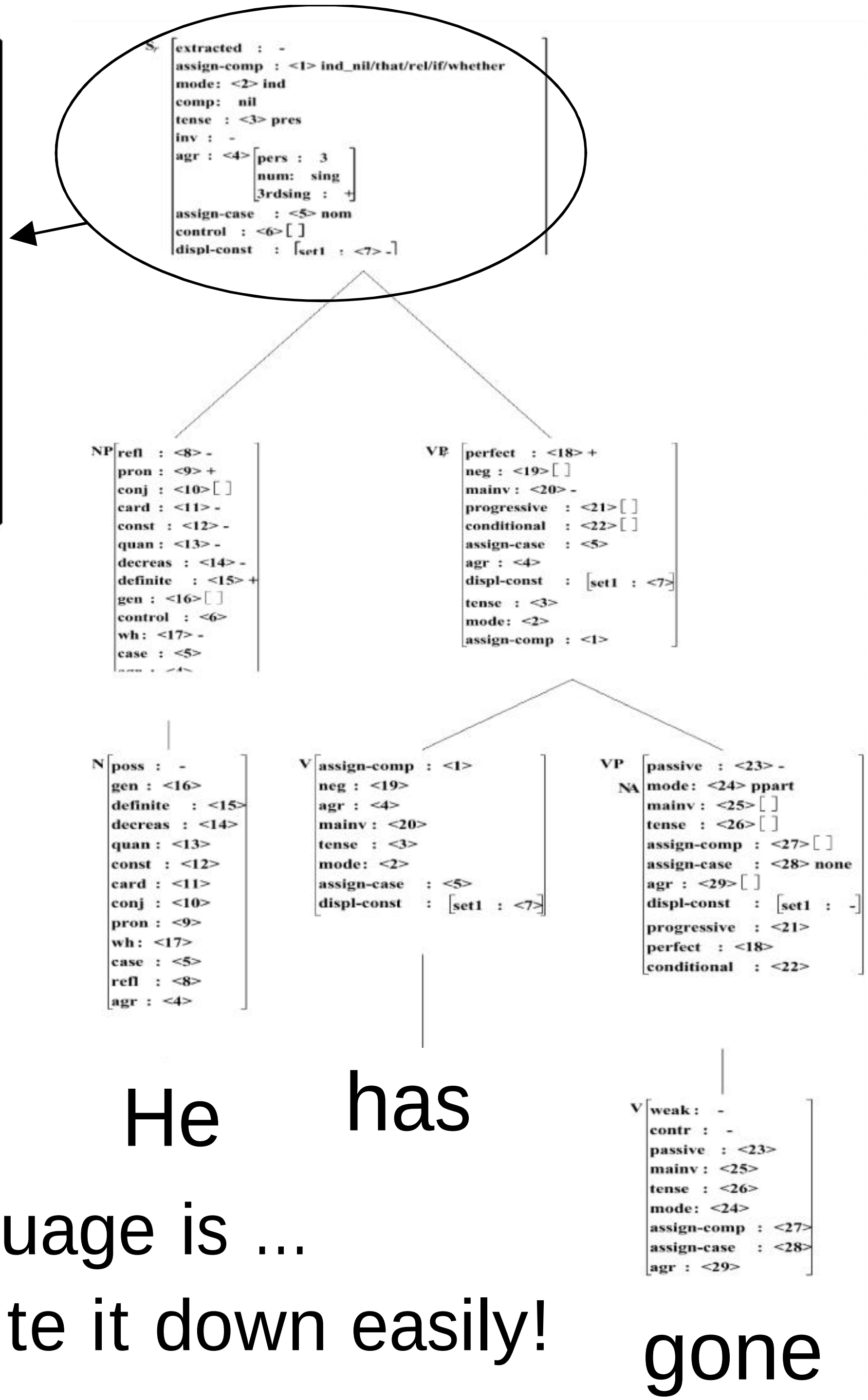


- Each phrase has a unique index such as "i".
- In some theories, coindexation is used to help extract a meaning from the tree.
- In other theories, it is just an aid to help you follow the example.

the money$_i$ I spend e_i on the happiness$_j$ I hope to buy e_j

which violin$_i$ is this sonata$_j$ easy to play e_j on e_i





- Lots of attributes
  (tense, number, person, gaps, vowels, commas, wh, etc ...)

- Sorry, that's just how language is ...
- You know too much to write it down easily!