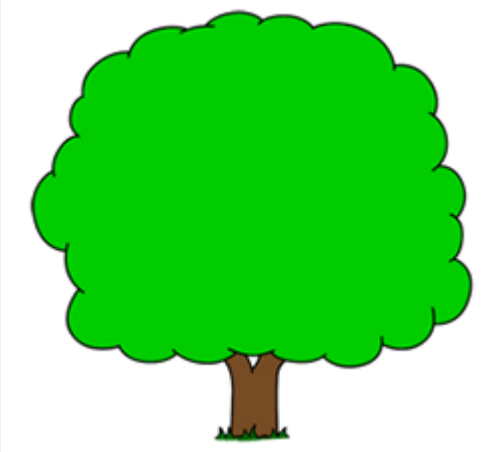
Gardener's advice: how to grow parse trees



Mariana Romanyshyn
Computational Linguist
at Grammarly

Natural Language Processing

- Machine Translation
- Event Extraction
- Sentiment Analysis
- Error Correction
- Automatic Summarization
- Question Answering

Prerequisites

- Sentence Detection
- Tokenization
- Lemmatization/Stemming
- Part-of-speech Tagging
- Named Entity Recognition
- Parsing (Syntactic Analysis)
- Coreference Resolution
- Relationship Extraction
- Word Sense Disambiguation

For example… (1)

BadCompany Inc., a high-flying company, filed a suit for $1.5B against PoorCompany Corp. and their investor, GoodCompany & Co. The company will take them to court on May 16, 2014.

For example... (2)

=> Sentence Detection

BadCompany Inc., a high-flying company, filed a suit for $1.5B against PoorCompany Corp. and their investor, GoodCompany & Co.

The company will take them to court on May 16, 2014.

For example… (3)

=> Tokenization

[“BadCompany” “Inc.” “,” “a” “high-flying” “company” “,” “filed” “a” “suit” “for” “$” “1.5” “B” “against” “PoorCompany” “Corp.” “and” “their” “investor” “,” “GoodCompany” “&” “Co.” “The” “company” “will” “take” “them” “to” “court” “on” “May” “16” “,” “2014” “.”]

For example… (4)

=> Lemmatization

[“BadCompany” “Inc.” “,” “a” “high-flying” “company” “,” “file” “a” “suit” “for” “$” “1.5” “B” “against” “PoorCompany” “Corp.” “and” “their” “investor” “,” “GoodCompany” “&” “Co.” “The” “company” “will” “take” “them” “to” “court” “on” “May” “16” “,” “2014” “.”]

For example… (5)

=> POS tagging

[“BadCompany”/NNP “Inc.”/NNP “,”/|,| “a”/DT “high-flying”/JJ “company”/NN “,”/|,| “filed”/VBD “a”/DT “suit”/NN “for”/IN “$”/$ “1.5”/CD “B”/CD “against”/IN “PoorCompany”/NNP “Corp.”/NNP “and”/CC “their”/PRP$ “investor”/NN “,”/|,| “GoodCompany”/NNP “&”/CC “Co.”/NNP]

[“The”/DT “company”/NN “will”/MD “take”/VB “them”/PRP “to”/TO “court”/NN “on”/IN “May”/NNP “16”/CD “,”/|,| “2014”/CD “.”/|.|]

For example… (6)

=> Named Entity Recognition

[“BadCompany”/NNP “Inc.”/NNP “,”/|,| “a”/DT “high-flying”/JJ “company”/NN “,”/|,| “filed”/VBD “a”/DT “suit”/NN “for”/IN “$”/$ “1.5”/CD “B”/CD “against”/IN “PoorCompany”/NNP “Corp.”/NNP “and”/CC “their”/PRP$ “investor”/NN “,”/|,| “GoodCompany”/NNP “&”/CC “Co.”/NNP]

[“The”/DT “company”/NN “will”/MD “take”/VB “them”/PRP “to”/TO “court”/NN “on”/IN “May”/NNP “16”/CD “,”/|,| “2014”/CD “.”/|.|]

For example... (7)

=> Sentence Parsing

```
(TOP
  (S
    (NP (DT The) (NN company))
    (VP (MD will)
          (VP (VB take)
                (NP (PRP them))
                (PP (TO to)
                      (NP (NN court)))
                (PP (IN on)
                       (NP (NNP May) (CD 16) (, ,) (CD 2014)))))
    (. .)))
```

For example… (8)

=> Coreference resolution

BadCompany Inc., a high-flying company, filed a suit for $1.5B against PoorCompany Corp. and their investor, GoodCompany & Co. The company will take them to court on May 16, 2014.

BadCompany Inc. => their, company
PoorCompany Corp. => them
GoodCompany & Co. => them

For example... (9)

=> Relationship Extraction

BadCompany Inc., a high-flying company, filed a suit for $1.5B against PoorCompany Corp. and their investor, GoodCompany & Co. The company will take them to court on May 16, 2014.

Suit: suer - BadCompany Inc.
defendant - PoorCompany Corp.
defendant - GoodCompany & Co.
sum - $1.5B
date - May 16, 2014

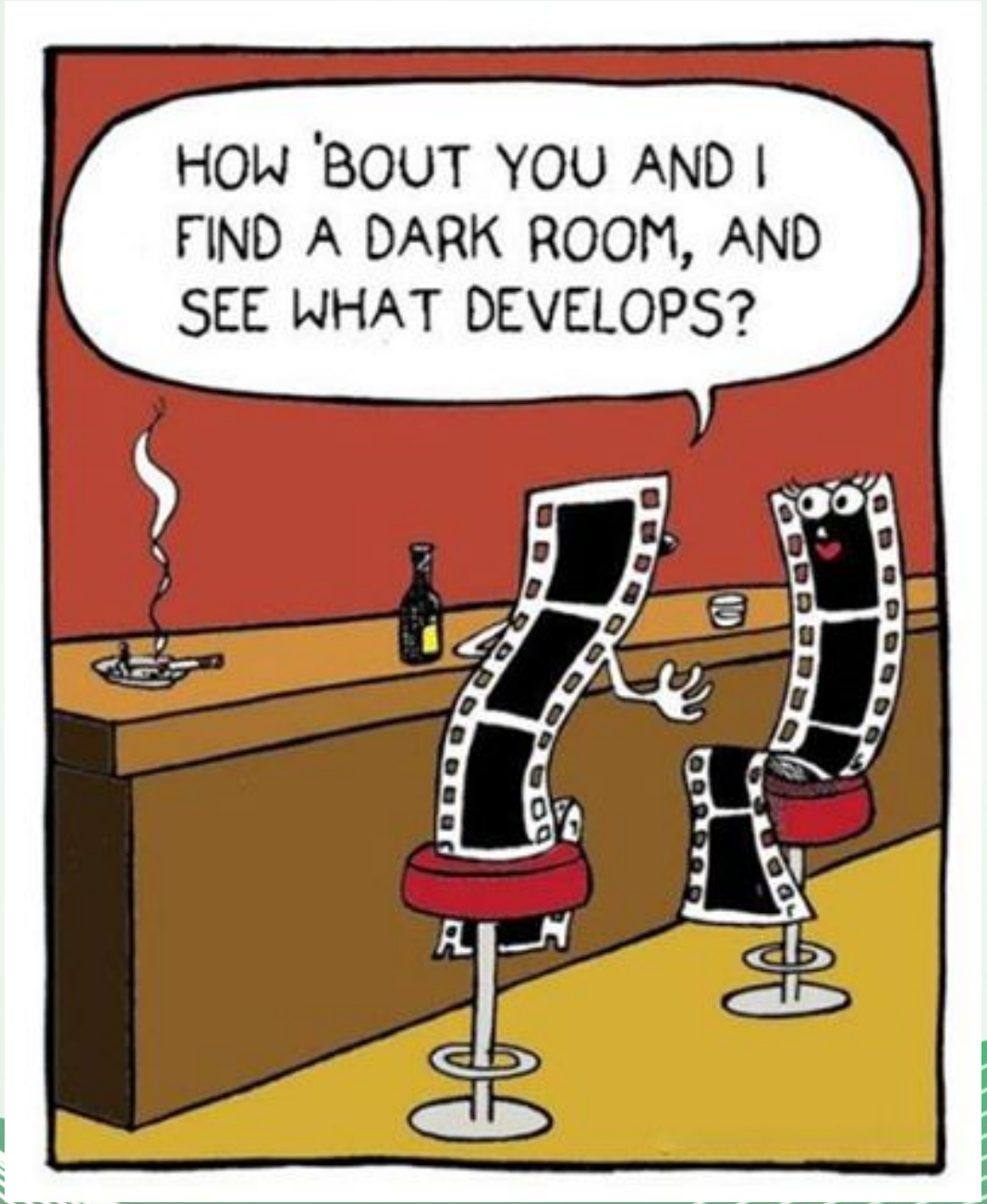
(investorOf: GoodCompany & Co., BadCompany Inc.)

For example... (10)

=> Word Sense Disambiguation

BadCompany Inc., a high-flying company, filed a suit for $1.5B against PoorCompany Corp. and their investor, GoodCompany & Co. The company will take them to court on May 16, 2014.

Suit, n:
- a set of clothes made from the same cloth
- a claim or complaint that someone makes in a court of law
- one of four sets of playing cards that together make a pack
- someone who works in an office and wears a suit



Sentence Parsing

Parsing – a method of understanding the meaning of a sentence.

Sentence Parsing

Parsing – a method of understanding the meaning of a sentence.

What's a sentence?

What's a sentence?

- Colorless green ideas sleep furiously.
- Furiously sleep ideas green colorless.

Noam Chomsky

What's a sentence?

- Colorless green ideas sleep furiously.
- Furiously sleep ideas green colorless.

Relations!

Noam Chomsky

Where do we get info on relations?

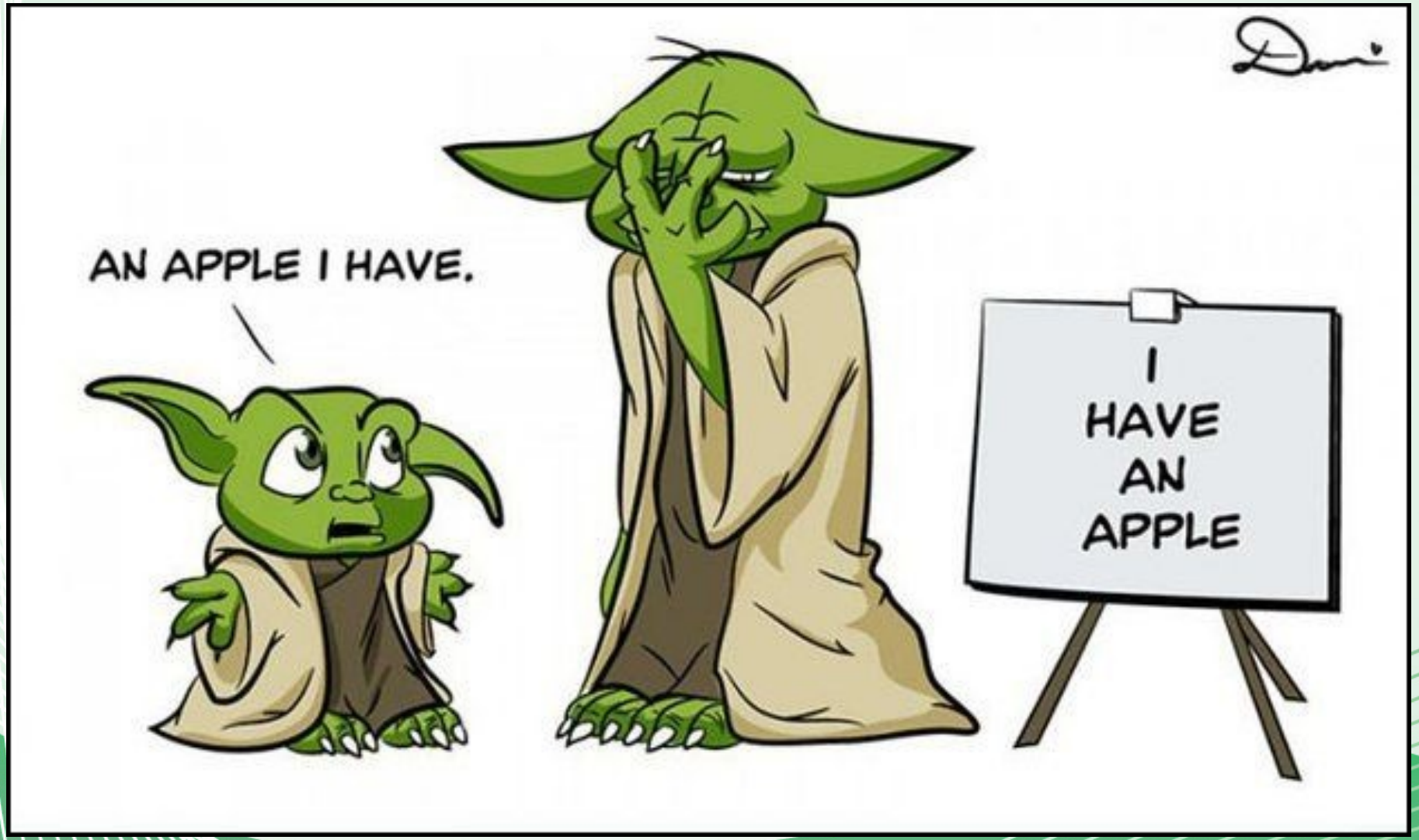
Languages: analytic or synthetic?

Analytic:

- word order
- additional words
- mostly uninflected

Synthetic:

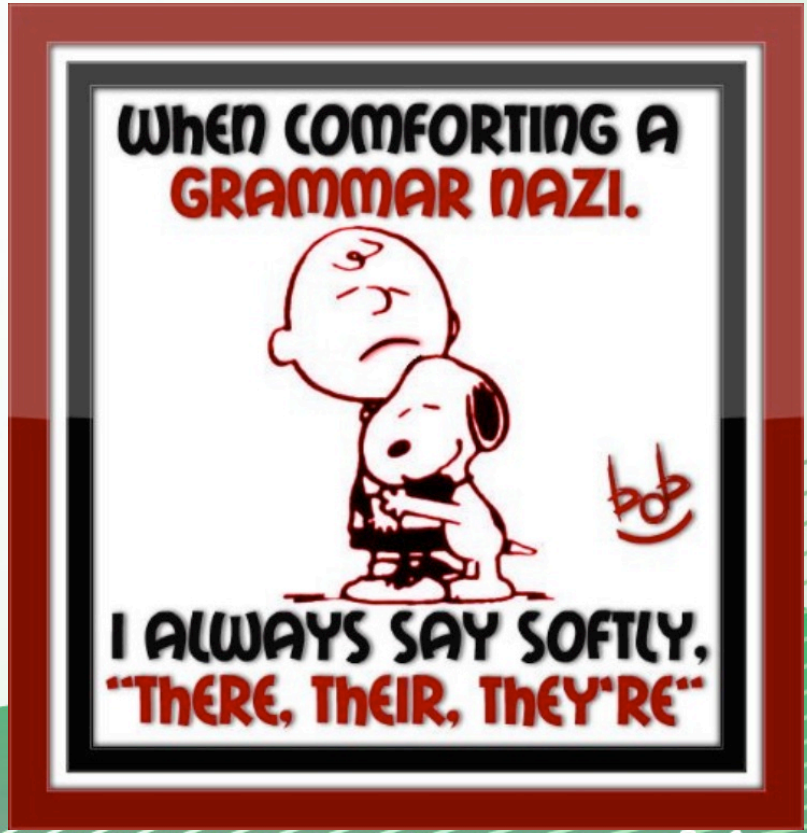
- lots of affixes
- word order is less important



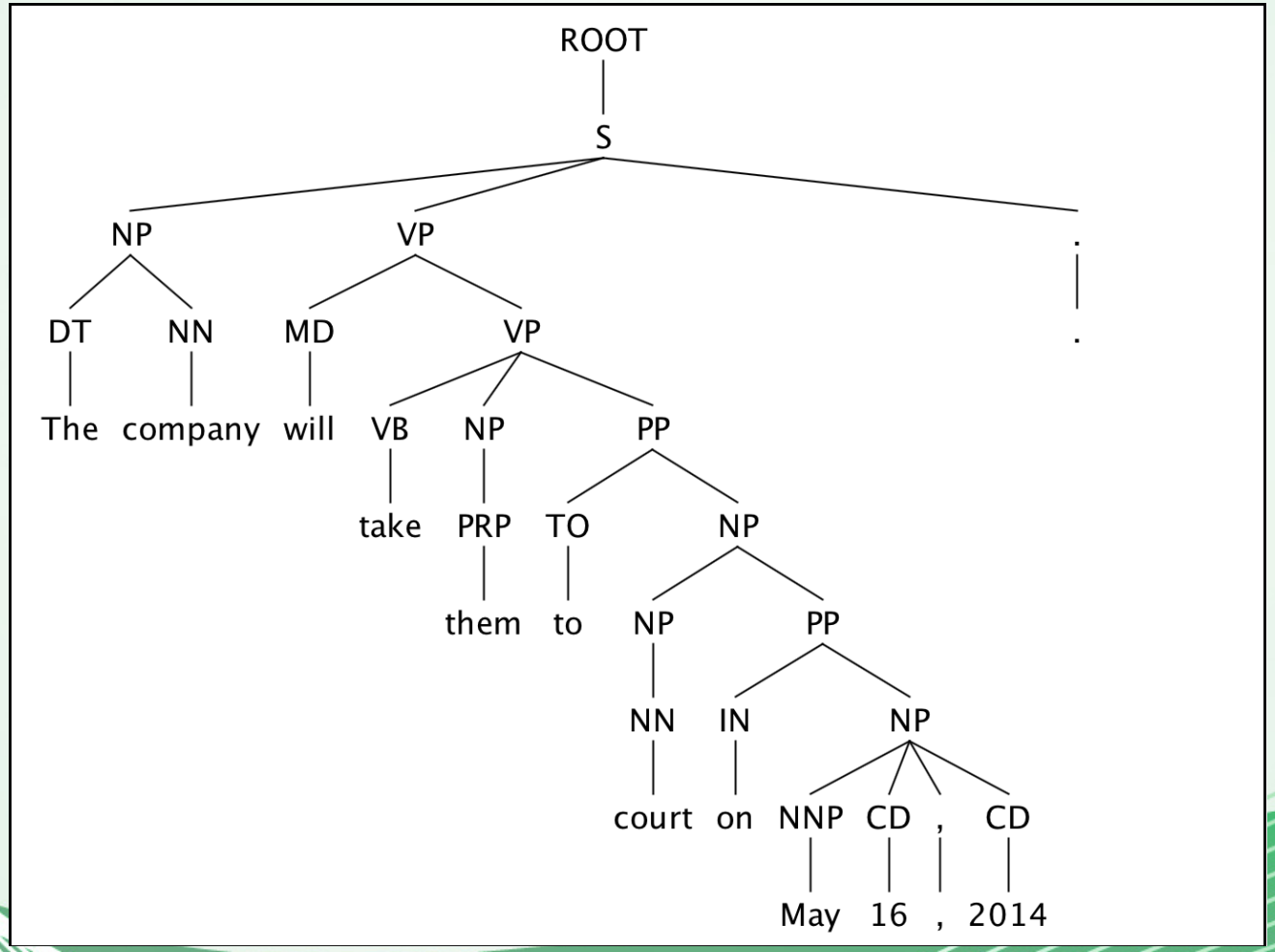
Types of Parsers

- Constituency Parser
- Dependency Parser

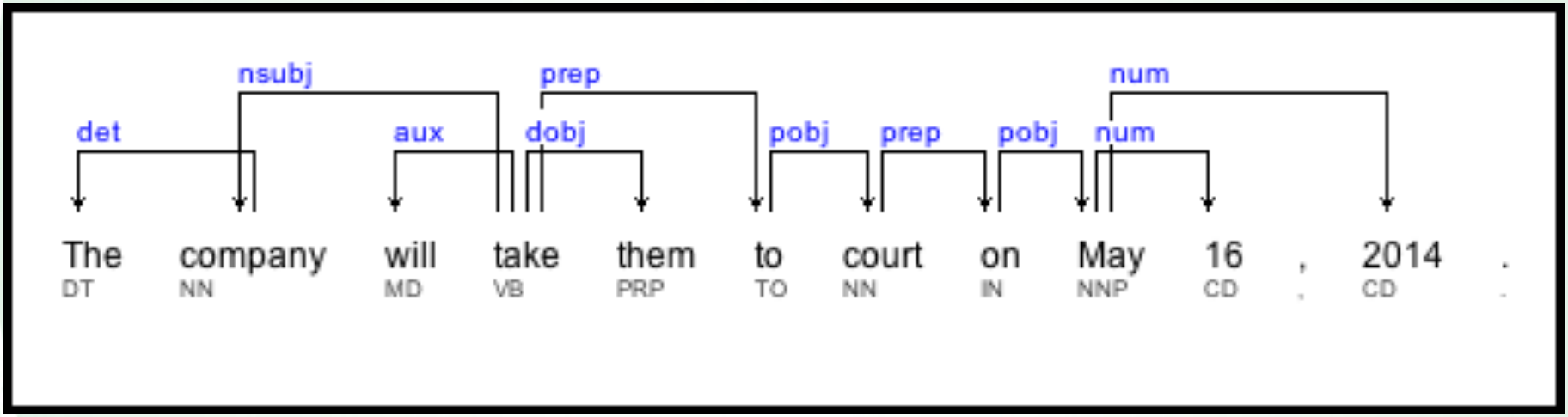
And both of them use GRAMMAR!



Constituency Parse Tree



Dependency Parse Tree



Context-Free Grammar

$G = (N, \sum, R, S)$, where

N – a final set of non-terminal symbols

{NP, VP, PP, S, SQ, SBAR, SBARQ …}

$\sum$ – a final set of terminal symbols

{NN, NNS, VB, VBZ, VBD, IN, TO, |,| …}

R – a finite set of rules

S – a start symbol for each tree (*TOP*/*ROOT*/*S1*)

Rules

$\alpha \rightarrow \beta$, where $\alpha \in N$ and $\beta \in (N \cup \sum)^+$

$S \rightarrow S\ CC\ S$

$S \rightarrow NP\ VP\ |.|$

$S \rightarrow NP\ VP$

$NP \rightarrow NP\ SBAR$

$NP \rightarrow NP\ PP$

$NP \rightarrow NN\ NN$

$NP \rightarrow NN$

$NP \rightarrow DT\ NN$

$VP \rightarrow VBP\ VP$

$VP \rightarrow VBZ\ PP$

$VP \rightarrow VBD\ NP$

$VP \rightarrow VBN$

$VP \rightarrow VBZ$

$VP \rightarrow VB$

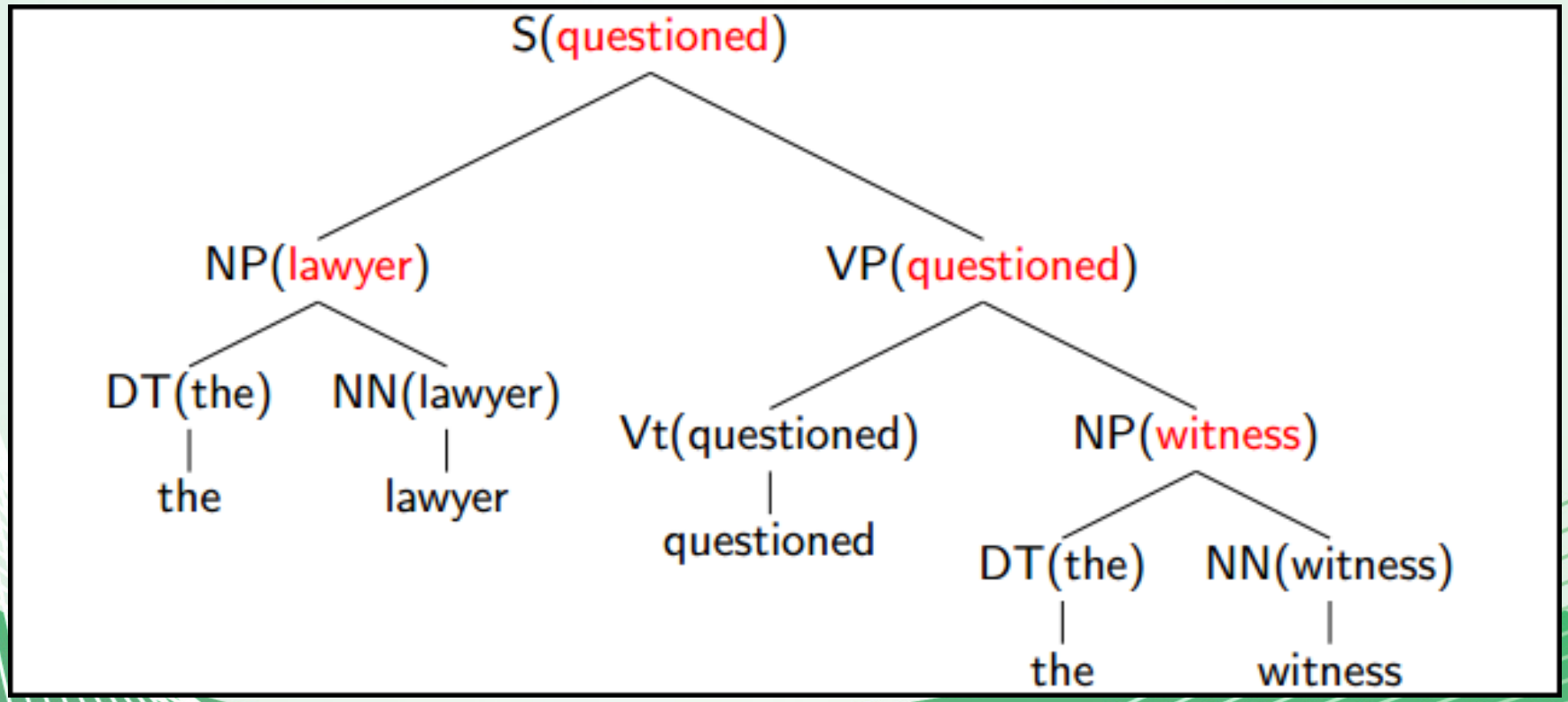
Probabilistic Context-Free Grammar

Probability of a tree: $p(t) = \prod_{i=1}^{n} q(\alpha_i \rightarrow \beta_i)$

S -> S CC S	*[0.2]*	*NP -> DT NN*	*[0.3]*
S -> NP VP |.|	*[0.6]*	*VP -> VBP VP*	*[0.2]*
S -> NP VP	*[0.2]*	*VP -> VBZ PP*	*[0.1]*
NP -> NP SBAR	*[0.1]*	*VP -> VBD NP*	*[0.1]*
NP -> NP PP	*[0.3]*	*VP -> VBN*	*[0.2]*
NP -> NN NN	*[0.15]*	*VP -> VBZ*	*[0.3]*
NP -> NN	*[0.15]*	*VP -> VB*	*[0.1]*

Lexicalized Grammars

Every rule has one special child – its head.



Context-Sensitive Grammar

Rules are of the form:

$$\alpha A \beta \text{ -> } \alpha \gamma \beta,$$

where:

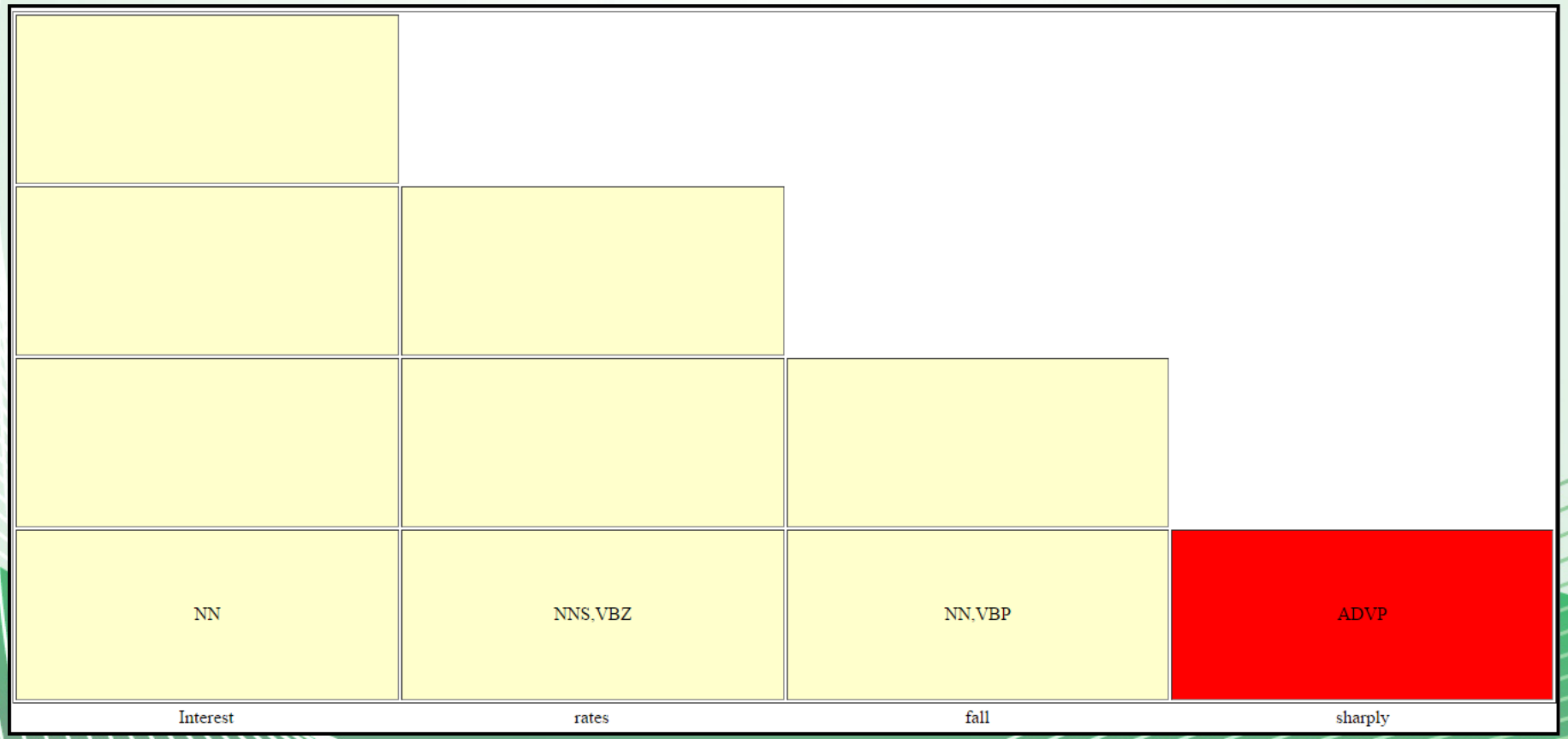
- $A \in N$
- $\gamma \in (N \cup \sum)^{+}$
- $\alpha, \beta \in (N \cup \sum)^{*}$

How do we process rules?

- CKY algorithm (bottom-up)
- Earley algorithm (top-down)
- GLR algorithm (bottom-up)
- Recursive ascent algorithm (bottom-up)
- Recursive descent algorithm (top-down)
- Etc…

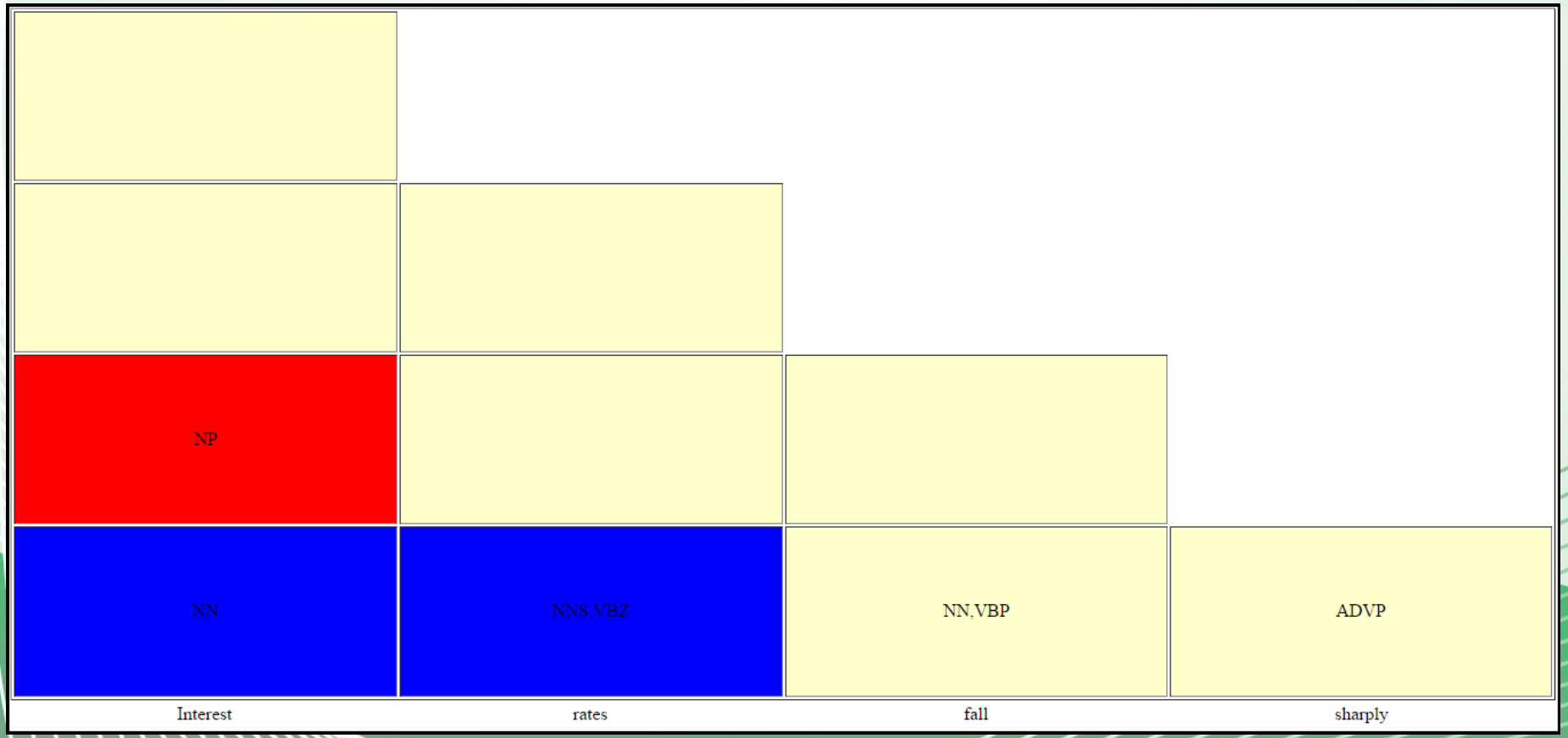
CKY (the Cocke–Kasami–Younger algorithm)

Sentence: *Interest rates fall by 5 points*



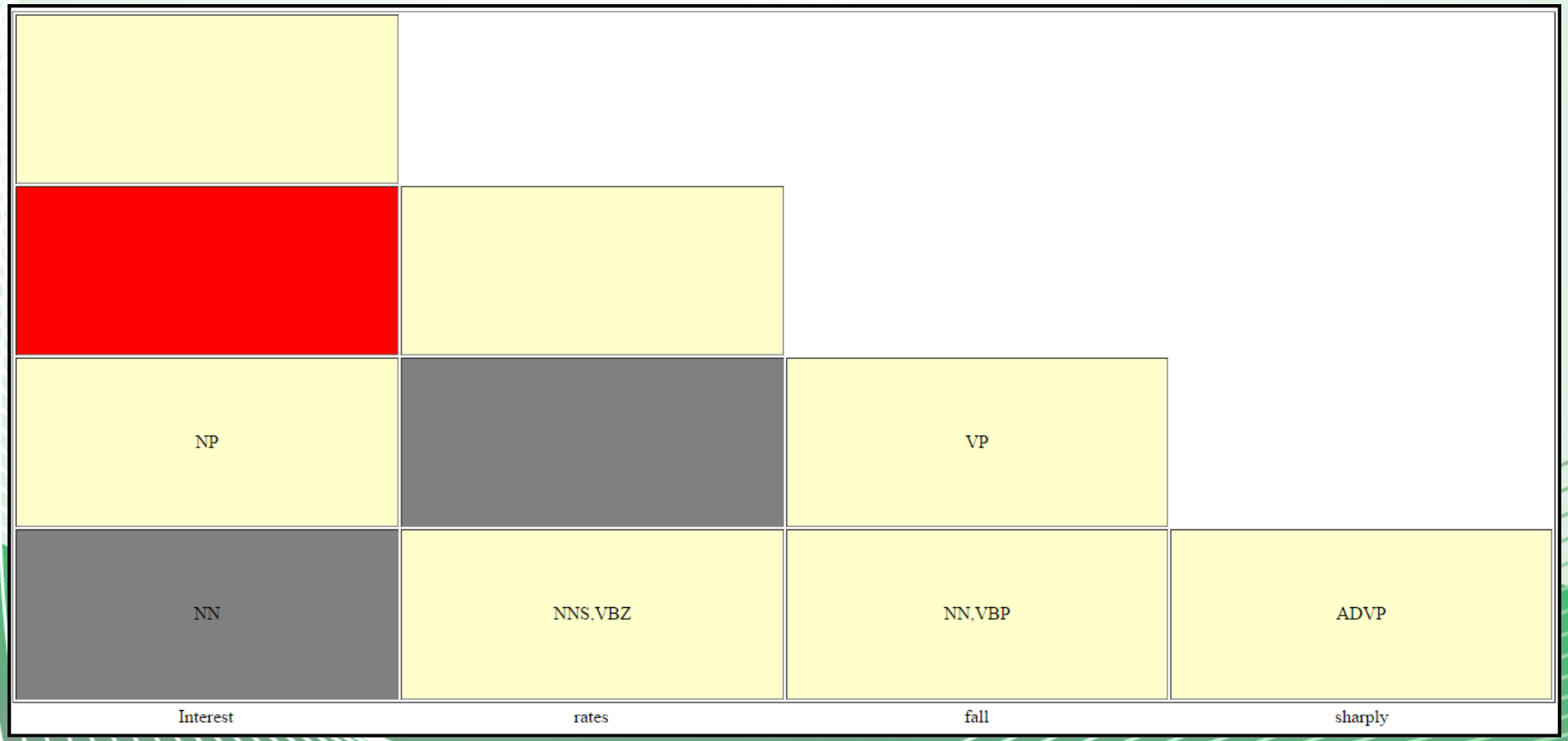
CKY (the Cocke–Kasami–Younger algorithm)

Sentence: *Interest rates fall by 5 points*



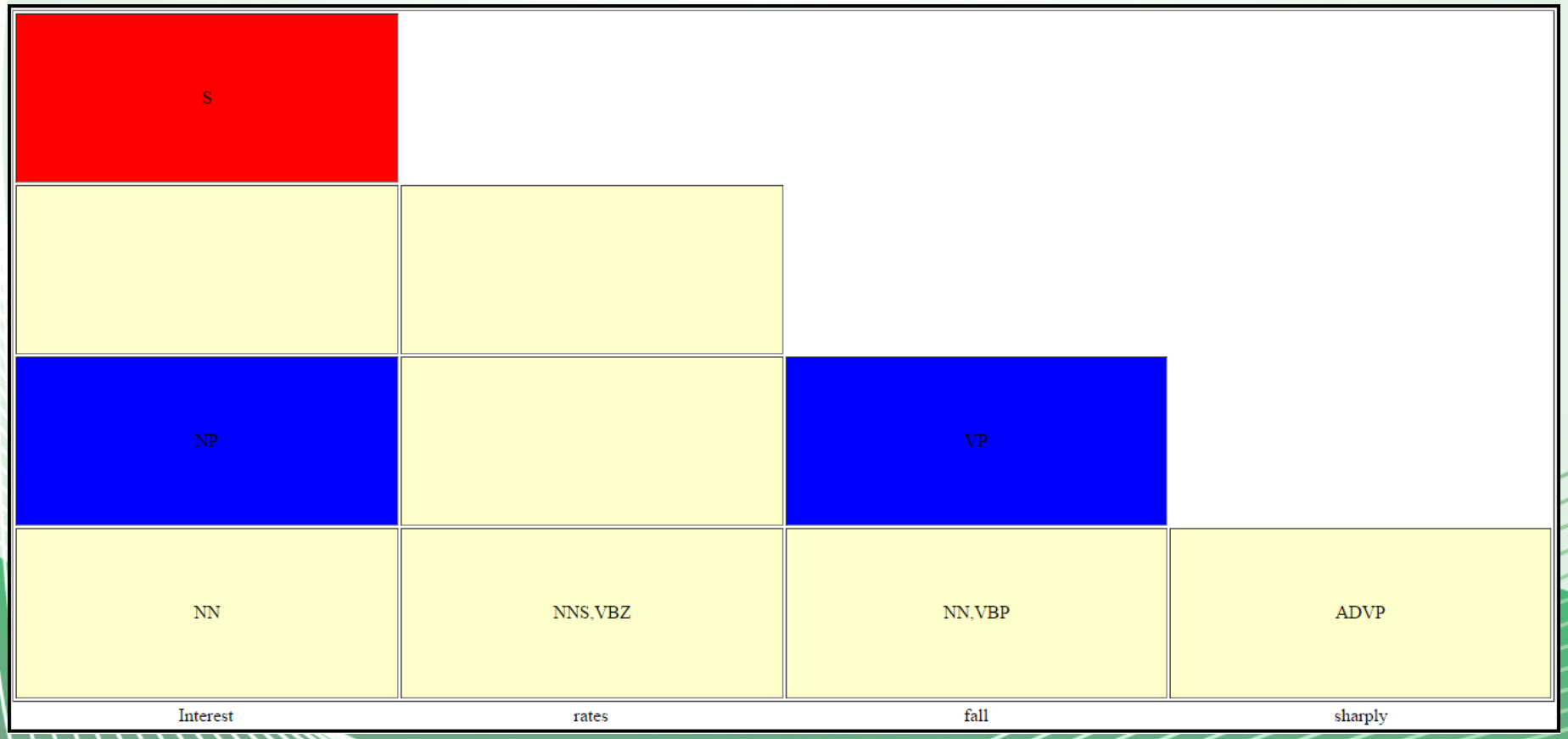
CKY (the Cocke–Kasami–Younger algorithm)

Sentence: *Interest rates fall by 5 points*



CKY (the Cocke–Kasami–Younger algorithm)

Sentence: *Interest rates fall by 5 points*



CKY (the Cocke–Kasami–Younger algorithm)

$\Theta(n^3 * G)$, where

- n – length of the string
- G – no. of rules

Uses Chomsky Normal Form:

$$A \rightarrow a \quad \text{or} \quad A \rightarrow B\ C,$$

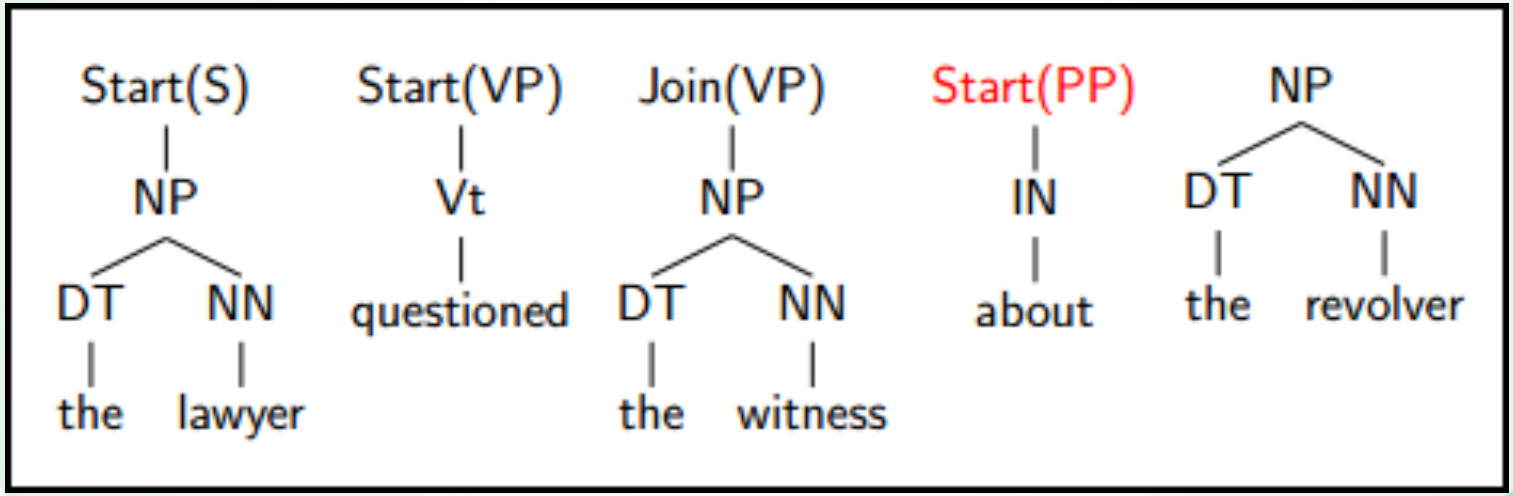
where

- $A, B, C \in N$
- $a \in \sum$

What about grammar-free parsers?

History-based models:

a tree is a sequence of decisions



The Most Famous Parsers

- BUBS (35 sents/sec)
- Zpar (24 sents/sec)
- OpenNLP (16 sents/sec)
- Berkeley (3.8 sents/sec)
- Stanford (2.3 sents/sec)
- Charniak (1.7 sents/sec)
- Enju (1.1 sents/sec)

Speed wise!

The Most Famous Parsers

- Zpar (~89%)
- Berkeley (~88%)
- OpenNLP (~88%)
- Charniak (~87%)
- Stanford (~86%)
- Enju (~86%)
- BUBS (~83%)

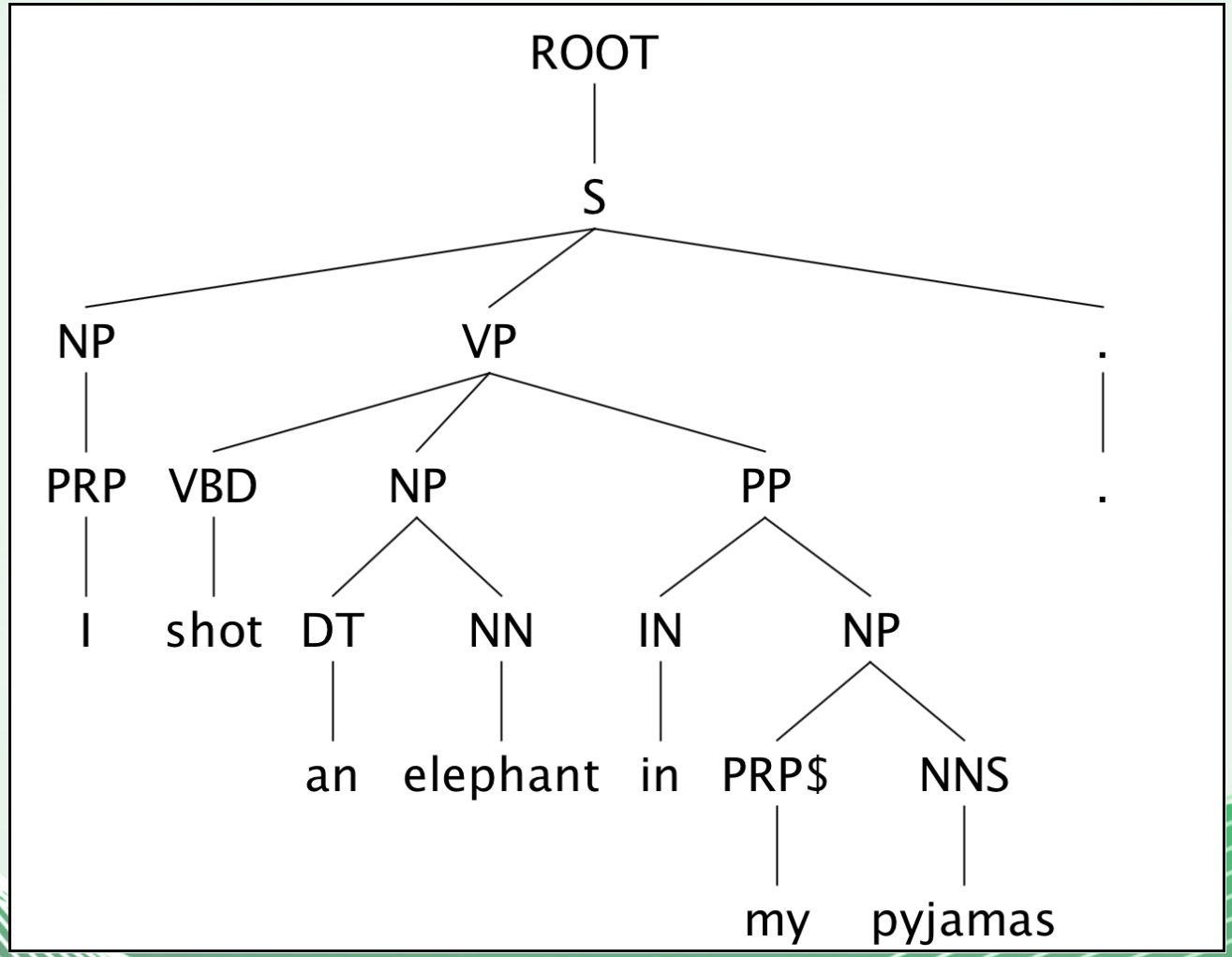
Accuracy wise!

Tricky Cases

I shot an elephant in my pyjamas.

P. S. How he got into my pyjamas I'll never know.

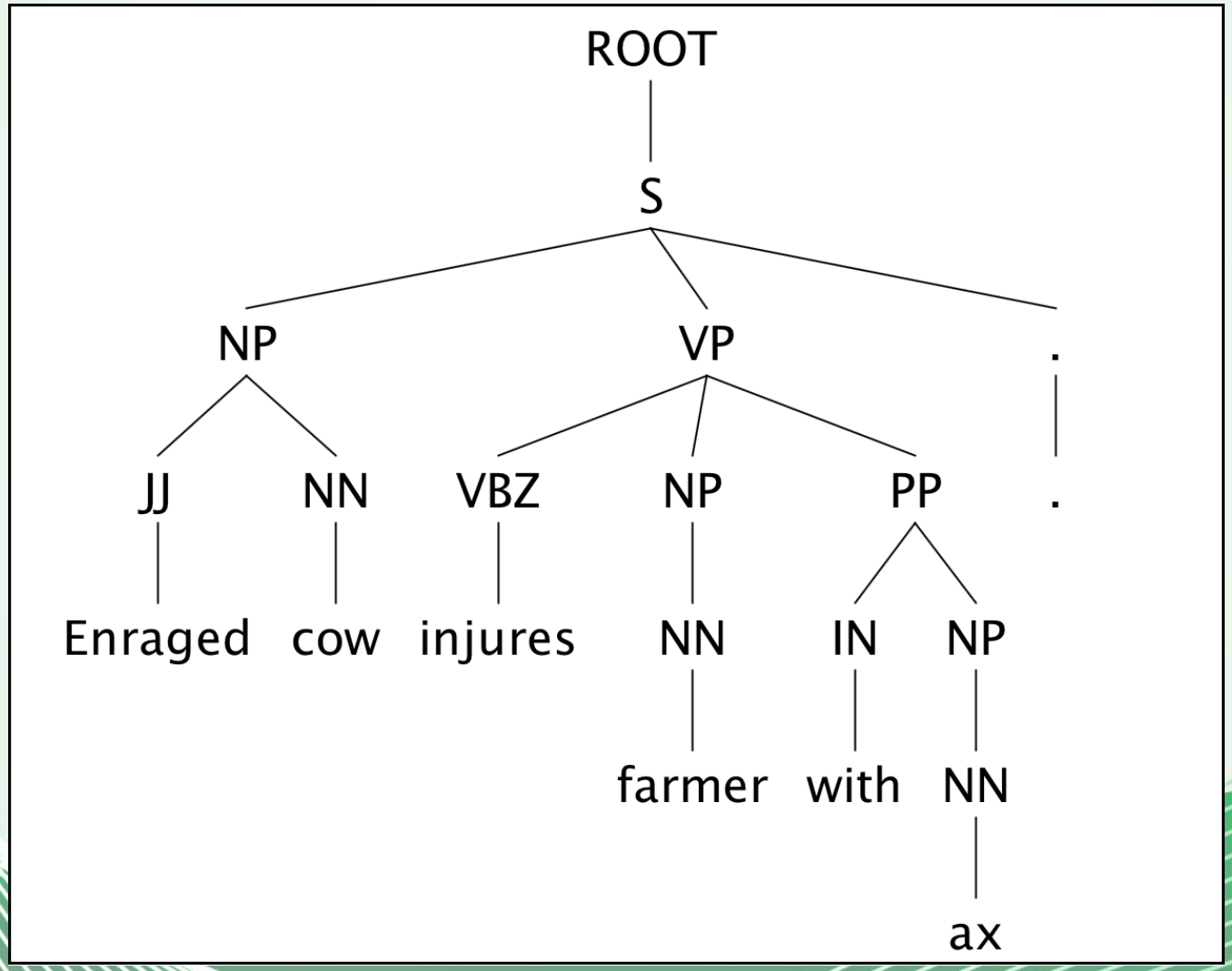
Stanford says…



Tricky Cases

Enraged cow injures farmer with ax.

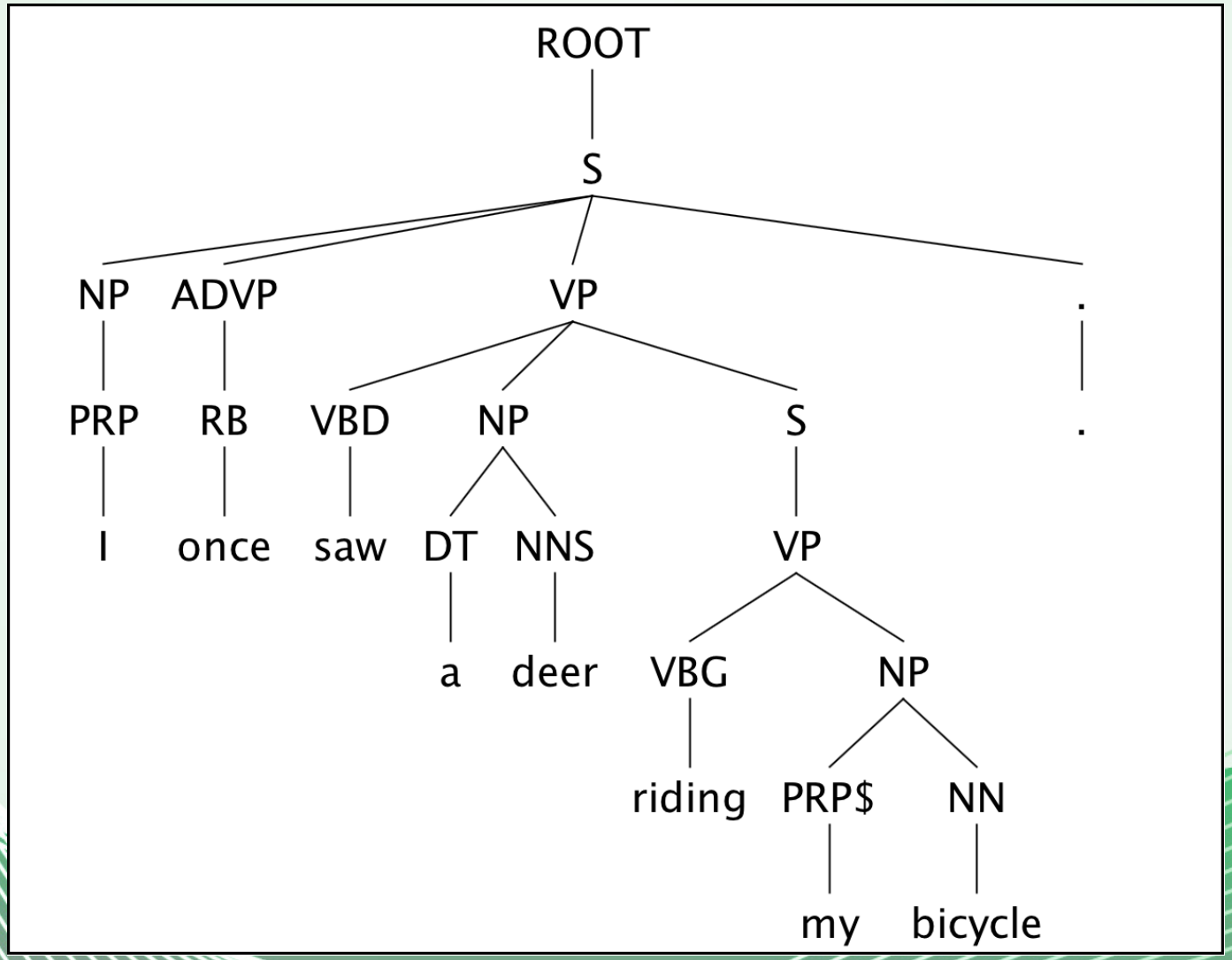
Stanford says…



Tricky Cases

I once saw a deer riding my bicycle.

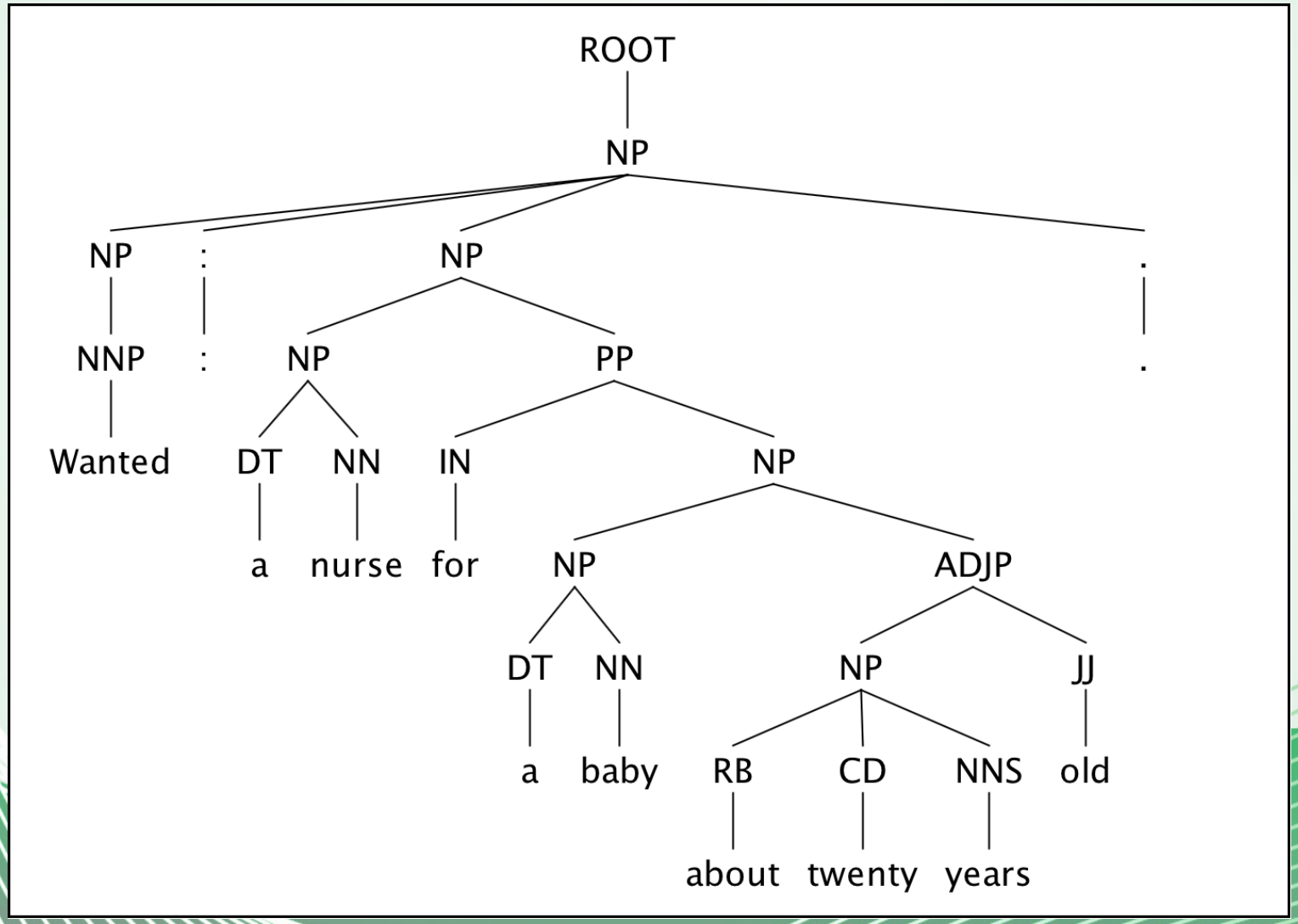
Stanford says…



Tricky Cases

Wanted: a nurse for a baby about twenty years old.

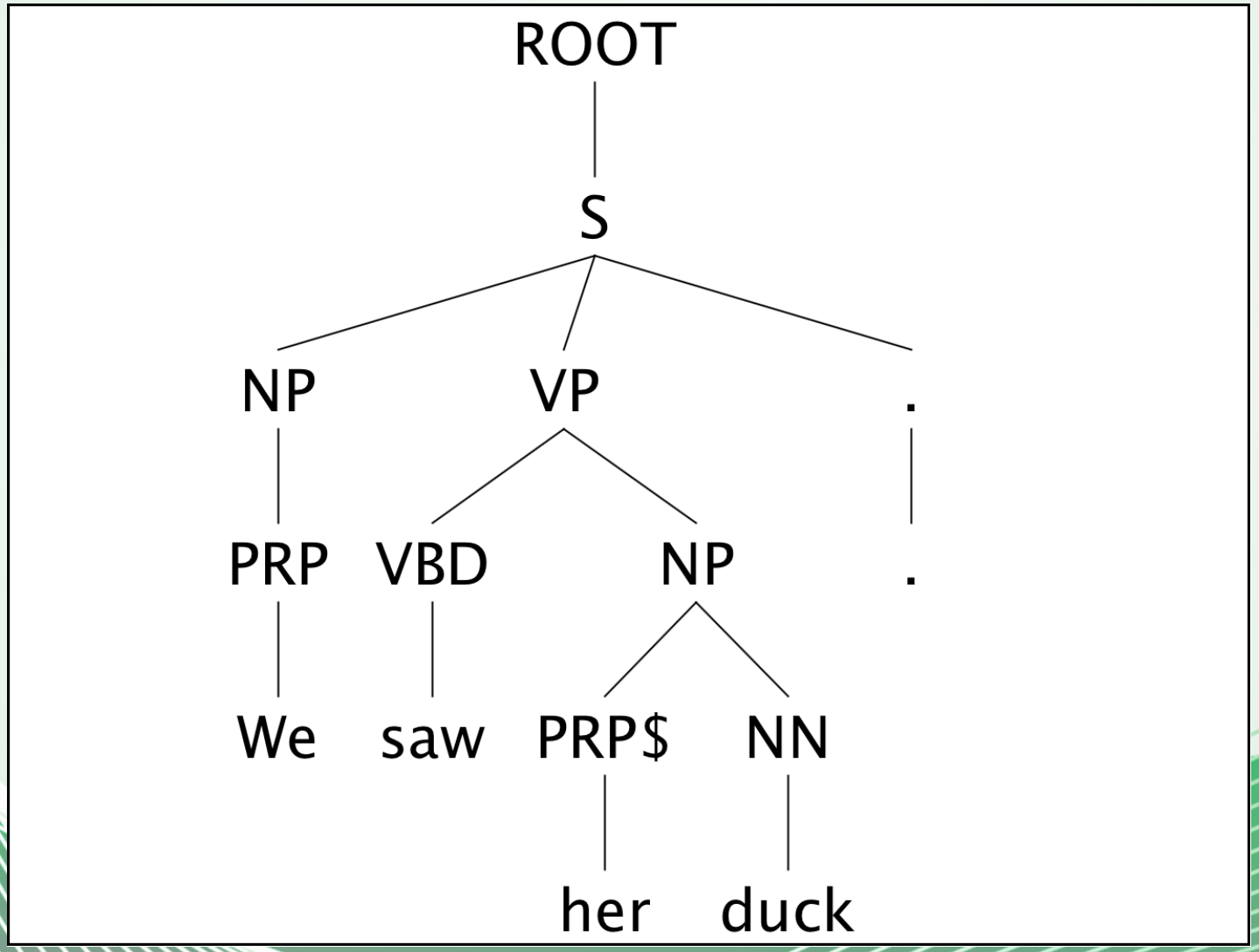
Stanford says…



Tricky Cases

We saw her duck.

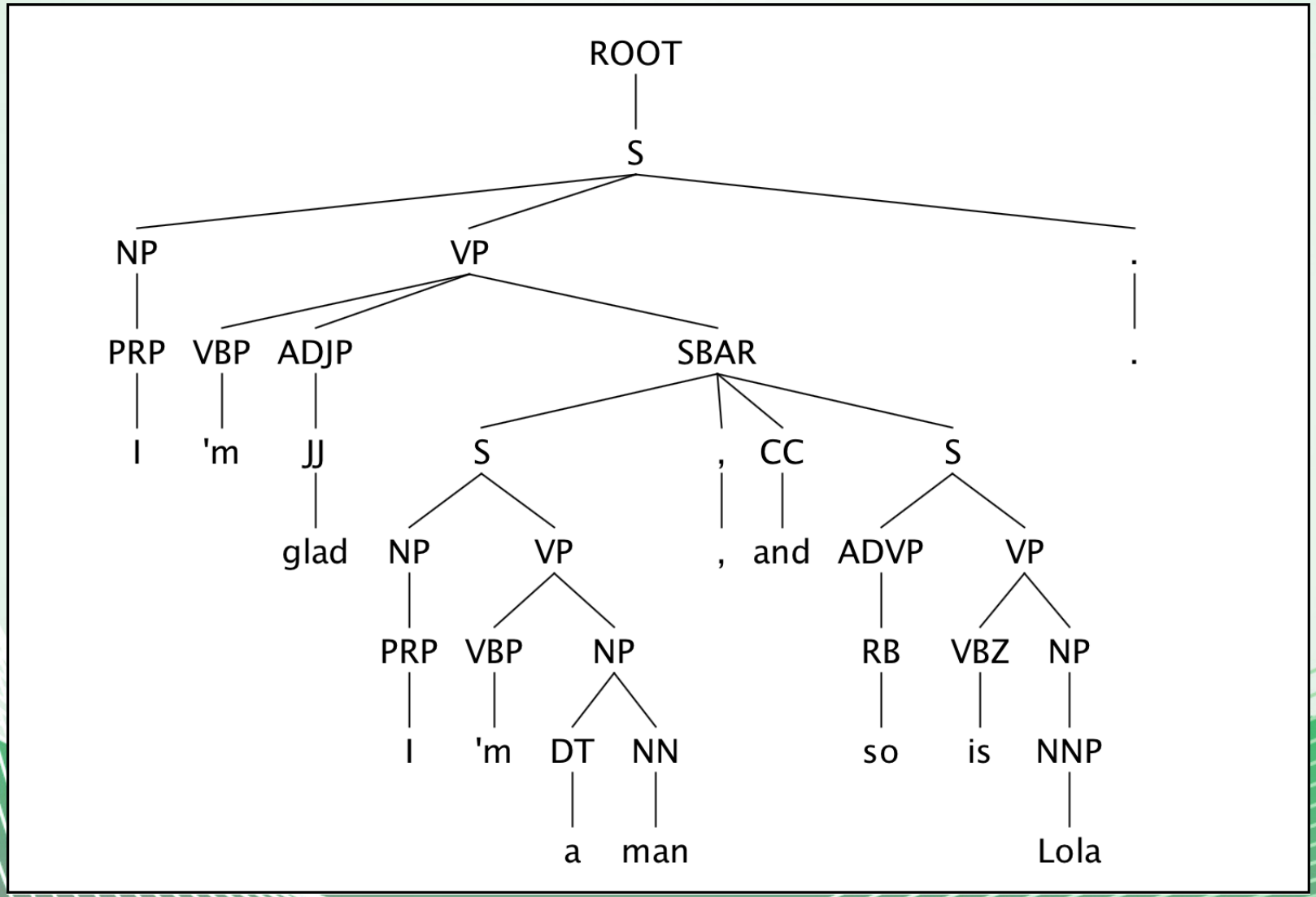
Stanford says…



Tricky Cases

I'm glad I'm a man, and so is Lola.

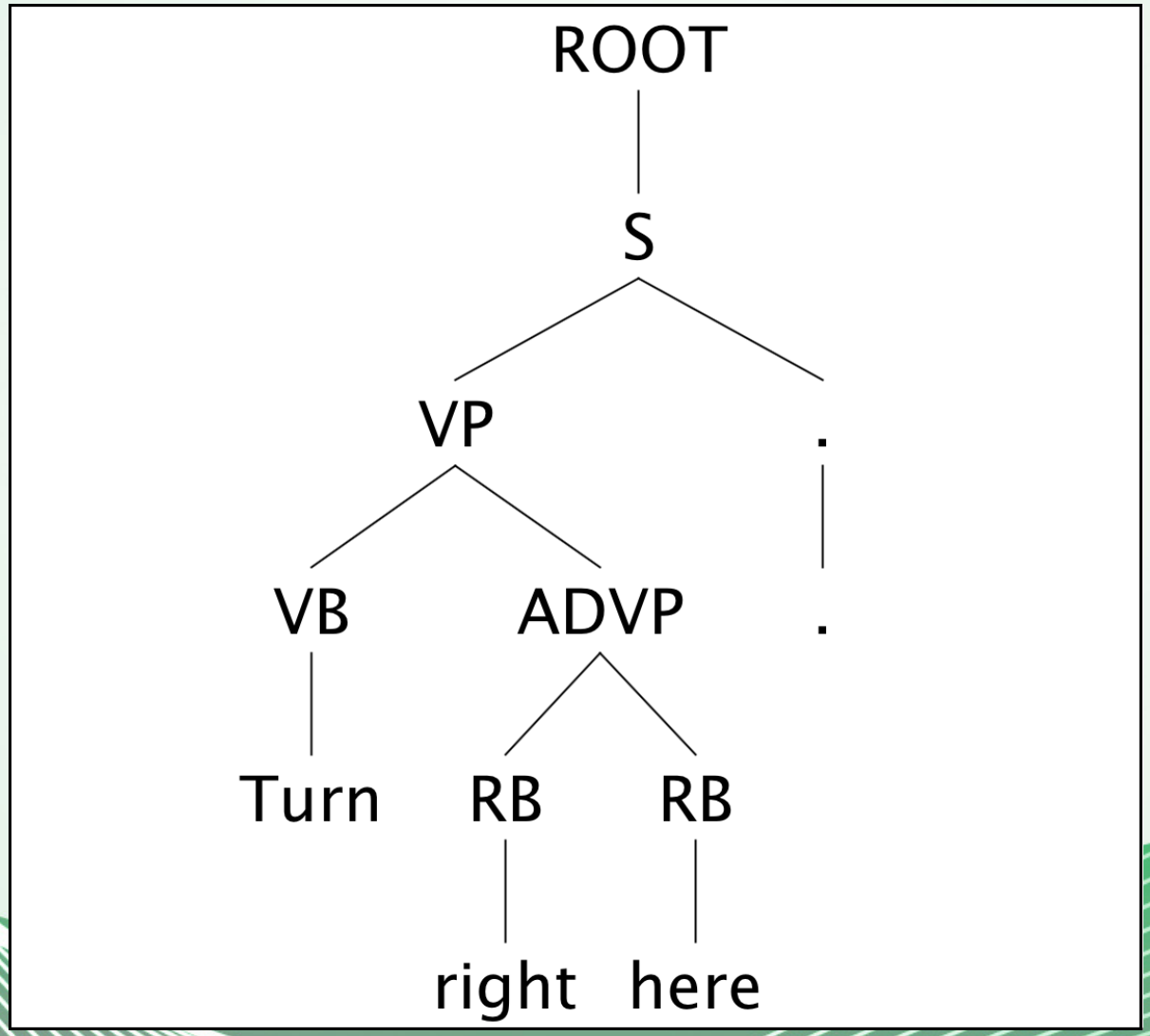
Stanford says...



Tricky Cases

Turn right here.

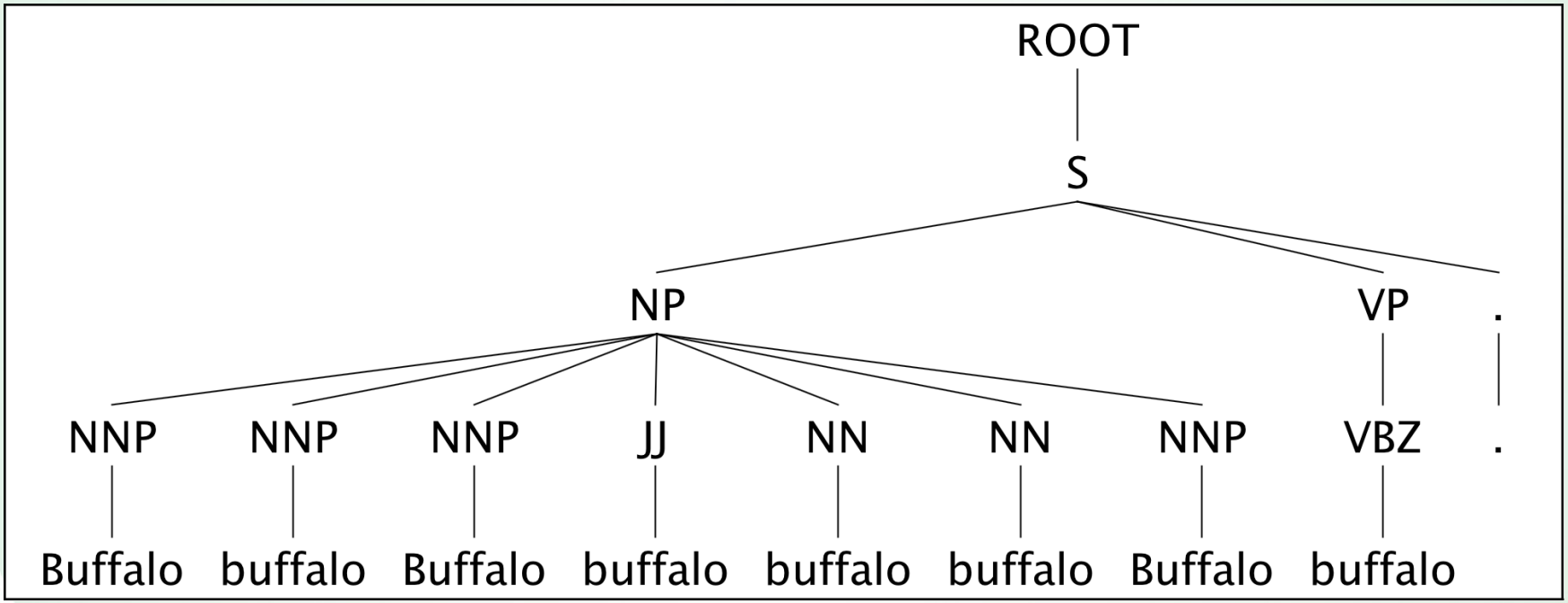
Stanford says…



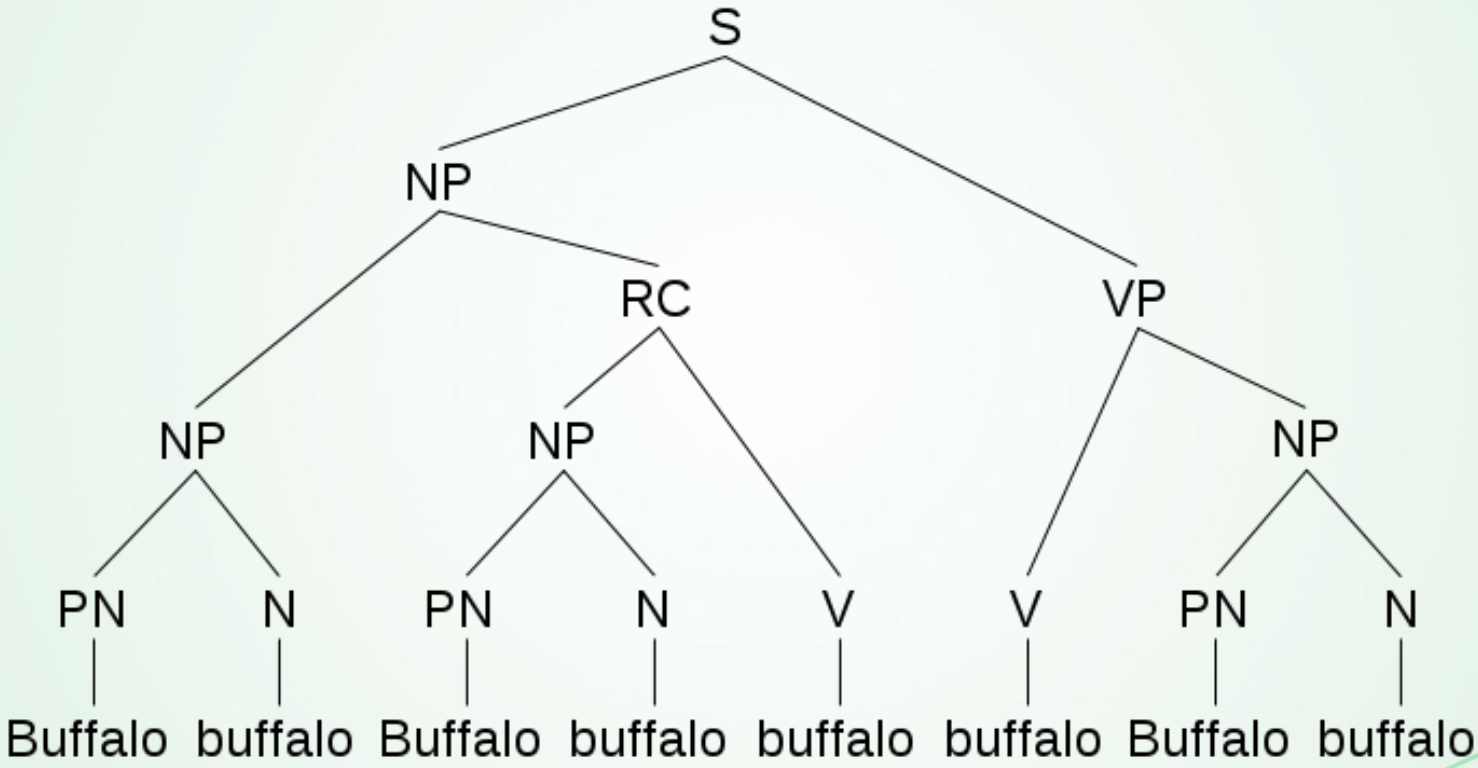
Tricky Cases

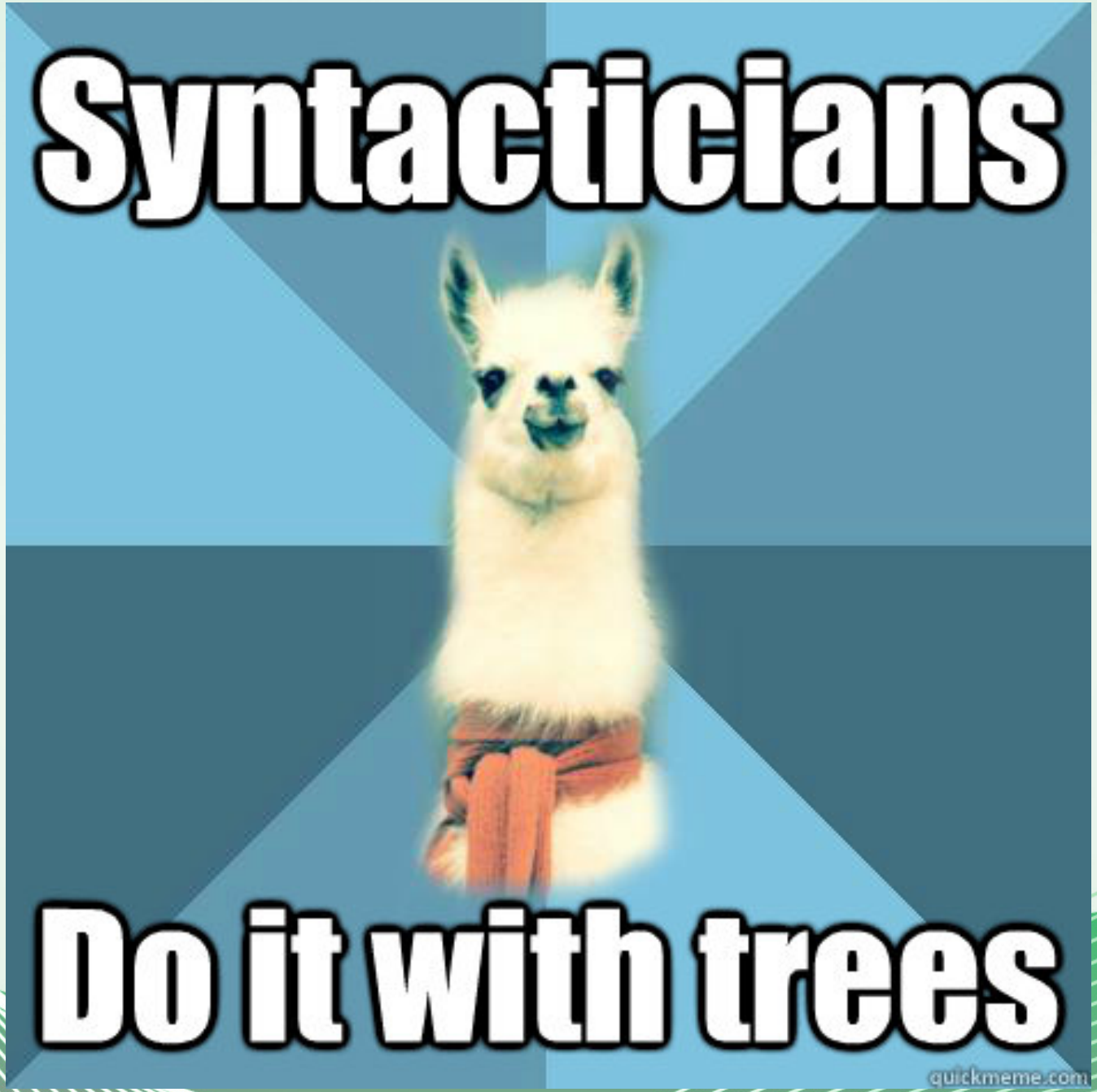
Buffalo buffalo Buffalo buffalo buffalo buffalo buffalo Buffalo buffalo.

Stanford says… WHAT?



What it should have been…





Questions?