LING/C SC 581:

Advanced Computational Linguistics

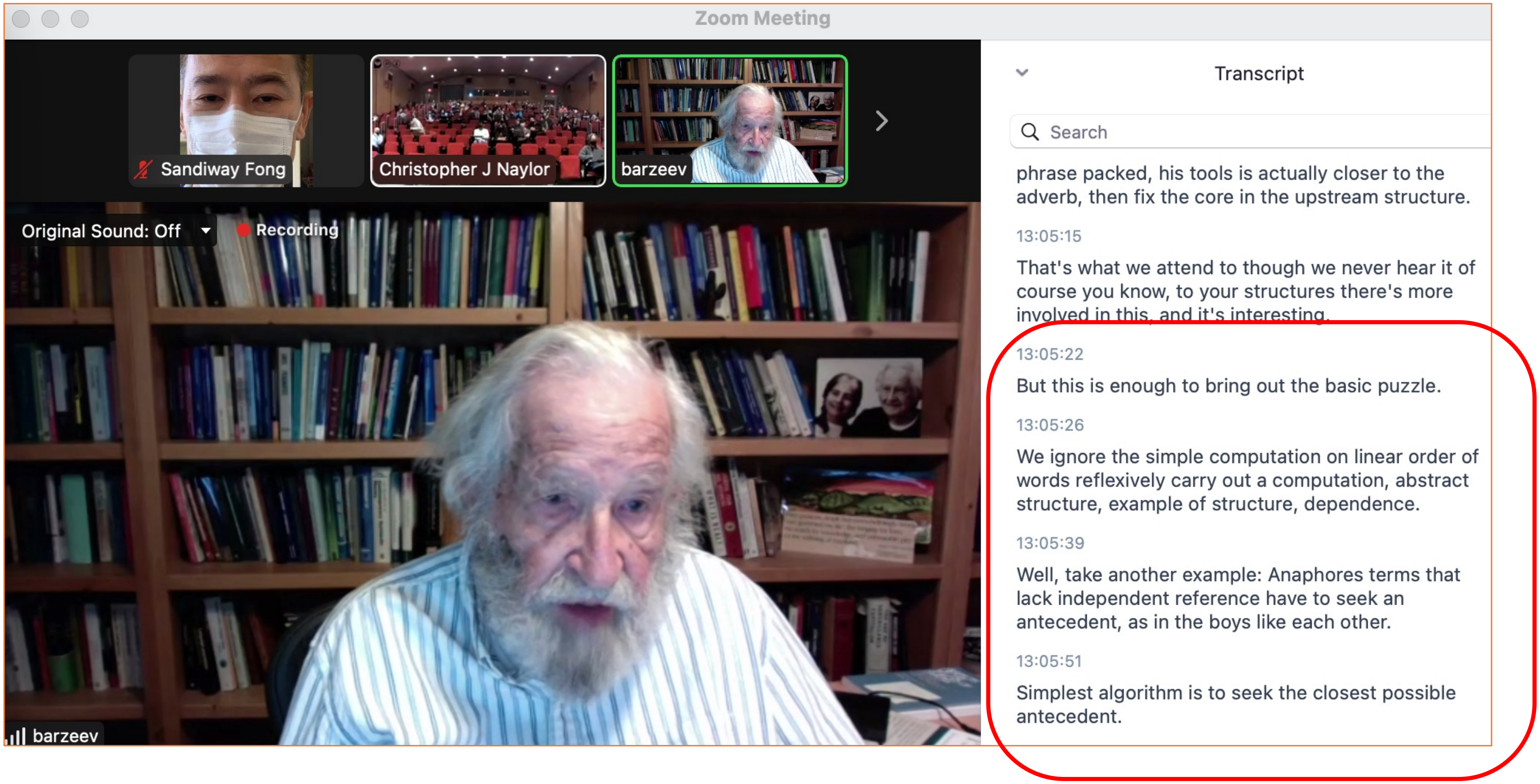
Lecture 22

Today's Topic

- Homework 10 Review
  - anaphor binding
  - some more *live* programming

- UA Prof. Chomsky gave a talk at MIT on Friday
  - he talked about *structure dependence* being fundamental to language
  - relevant to our homework
  - *also other examples* …

Last Friday's MIT Talk by UA Professor Noam Chomsky



phrase packed, his tools is actually closer to the adverb, then fix the core in the upstream structure.

13:05:15

That's what we attend to though we never hear it of course you know, to your structures there's more involved in this, and it's interesting.

13:05:22

But this is enough to bring out the basic puzzle.

13:05:26

We ignore the simple computation on linear order of words reflexively carry out a computation, abstract structure, example of structure, dependence.

13:05:39

Well, take another example: Anaphores terms that lack independent reference have to seek an antecedent, as in the boys like each other.

13:05:51

Simplest algorithm is to seek the closest possible antecedent.

Last Friday's MIT Talk by UA Professor Noam Chomsky

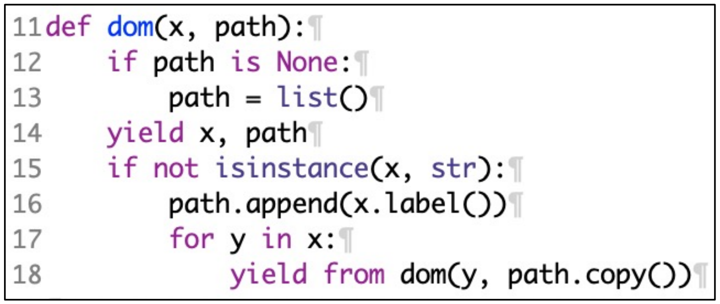
Homework 10: Question 1 Review

- Define appropriate global variables yregex and wregex to find candidate c-commanding NPs for anaphors ending in –*self*
- Search `ptb.parsed_sents()[70000:73451]`
- How many examples of NP c-commanding anaphors are there?
- 57

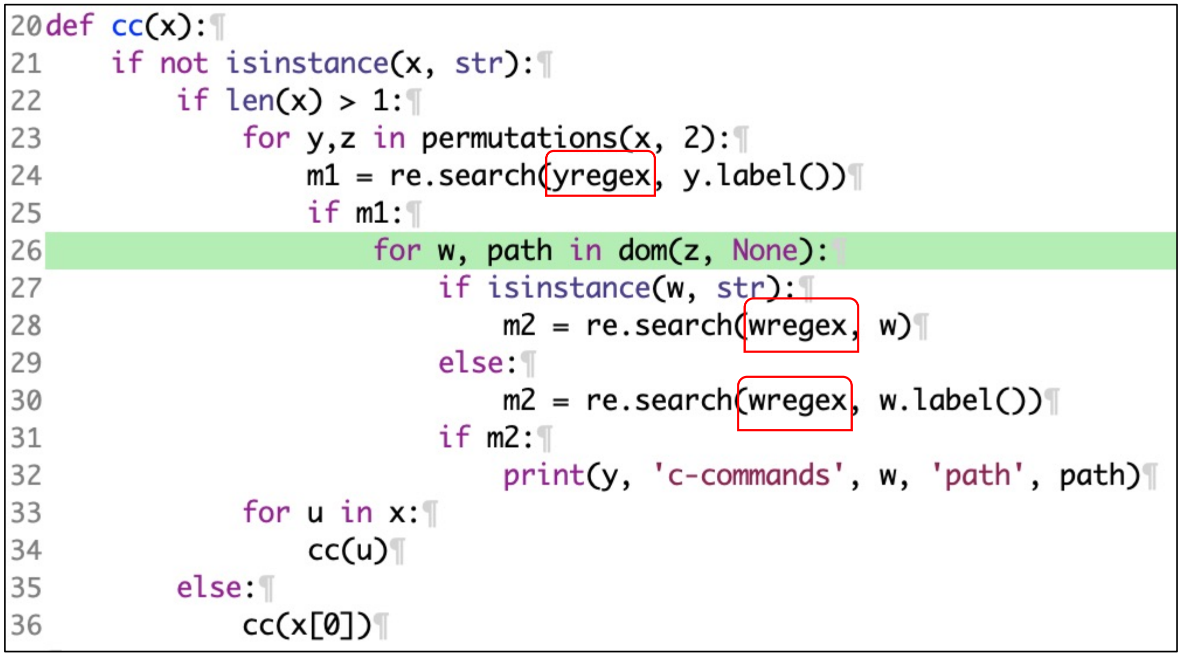
Homework 10: Question 1 Review

```
python -i ccommand3f.py
>>>
```

ccommand3f.py



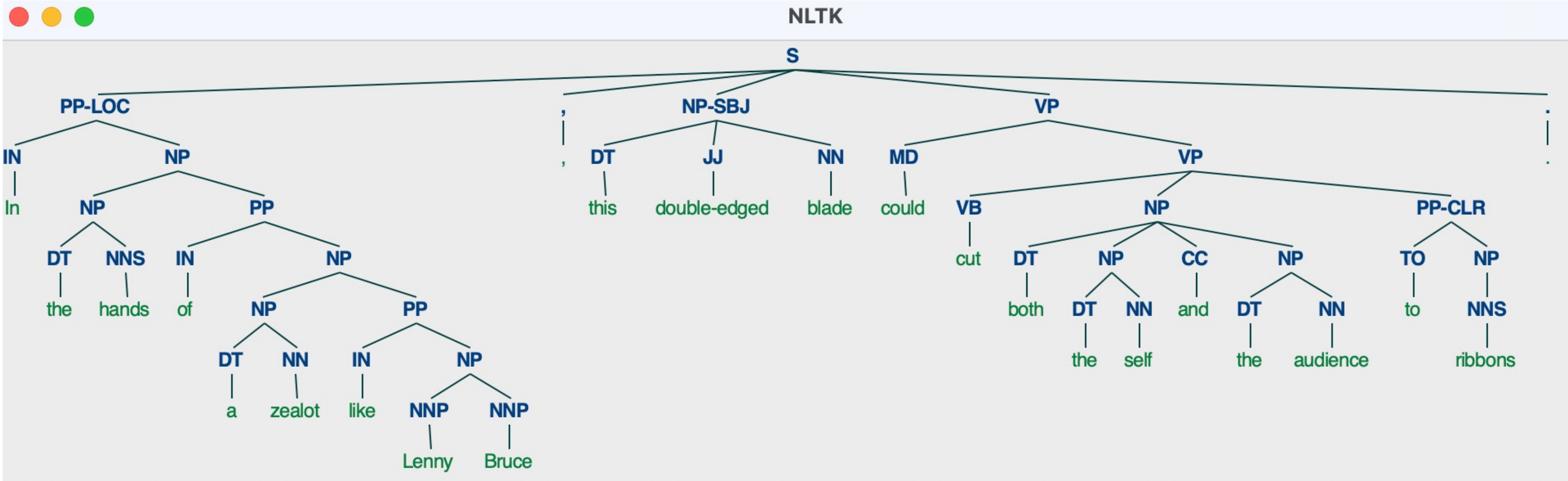
```python
def dom(x, path):
    if path is None:
        path = list()
    yield x, path
    if not isinstance(x, str):
        path.append(x.label())
        for y in x:
            yield from dom(y, path.copy())
```



```python
def cc(x):
    if not isinstance(x, str):
        if len(x) > 1:
            for y,z in permutations(x, 2):
                m1 = re.search(yregex, y.label())
                if m1:
                    for w, path in dom(z, None):
                        if isinstance(w, str):
                            m2 = re.search(wregex, w)
                        else:
                            m2 = re.search(wregex, w.label())
                        if m2:
                            print(y, 'c-commands', w, 'path', path)
        for u in x:
            cc(u)
    else:
        cc(x[0])
```

Homework 10: Question 1 Review

71206



Homework 10: Question 2 Review

- Search `ptb.parsed_sents()[70000:73451]` again
- How many anaphors have more than one candidate c-commanding NP in the same tree?

Homework 10: Question 2 Review

```
tree 70535 : (NP (NNP Little) (NNP Lily)) c-commands herself path ['SBAR',
'S', 'VP', 'S', 'NP-SBJ', 'PRP']

tree 70535 : (NP-SBJ (NNP Ms.) (NNP Cunningham)) c-commands herself path
['VP', 'S', 'NP-SBJ', 'PRP']

tree 70535 : (NP-PRD (-NONE- *?*)) c-commands herself path ['NP-SBJ',
'PRP']


tree 70709 : (NP-SBJ-1 (NNP Time)) c-commands itself path ['VP', 'S',
'VP', 'VP', 'NP', 'PRP']

tree 70709 : (NP-SBJ (-NONE- *-1)) c-commands itself path ['VP', 'VP',
'NP', 'PRP']
```

Homework 10: Question 2 Review

1. 70535: herself 3
2. 70709: itself 2
3. 70736: itself 2
4. 70740: itself 2
5. 70873: itself 2
6. 70935: itself 4
7. 71201: itself 3
8. 71229: itself 5
9. 71244: itself 6
10. 71466: itself 4
11. 71477: himself 4
12. 71709: itself 2
13. 71723: itself 3
14. 72266: itself 4
15. 72594: itself 4
16. 72668: himself 2

Homework 10: Question 2 Review

```
1. 70535 : 3
2. 70709 : 2
3. 70736 : 2
4. 70740 : 2
5. 70873 : 2
6. 70935 : 4
7. 71043 : 2
8. 71201 : 3
9. 71229 : 5
10.71244 : 6
11.71466 : 4
12.71477 : 4
13.71709 : 2
14.71723 : 3
15.71883 : 1
16.72266 : 4
17.72594 : 4
18.72668 : 2
19.72686 : 1
20.72709 : 1
```

Homework 10: Question 2 Review

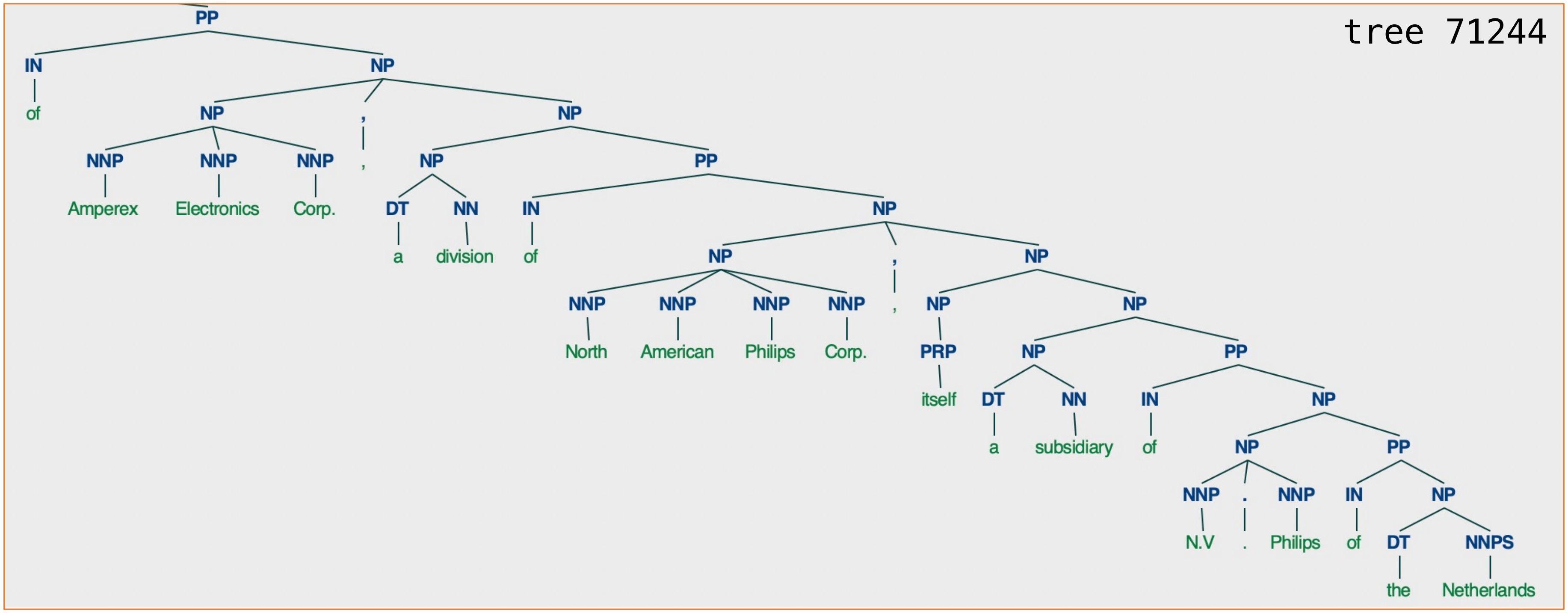
The quick-and-not-necessarily-accurate way

- First, record the tree number. Recall `enumerate()`?
- How to modify the code to print the numbers?
  - use a `total` counter
  - can also use a `subtotal` counter for each time cc(*tree*) is called
  - increment both counters if an appropriate c-commanding relations was found
  - if the `subtotal` counter was updated past 1, perhaps there was ambiguity, so print the tree number and the `subtotal` difference.

Homework 10: Question 2 Review

```
tree 71244 : (NP-SBJ (PRP He)) c-commands itself path ['VP', 'VP', 'NP-PRD', 'PP', 'NP', 'NP', 'PP', 'NP', 'NP', 'NP', 'PRP']
tree 71244 : (NP
  (NP (NN president))
  (CC and)
  (NP (JJ chief) (JJ executive) (NN officer))) c-commands itself path ['PP', 'NP', 'NP', 'PP', 'NP', 'NP', 'NP', 'PRP']
tree 71244 : (NP (NNP Amperex) (NNP Electronics) (NNP Corp.)) c-commands itself path ['NP', 'PP', 'NP', 'NP', 'NP', 'PRP']
tree 71244 : (NP (DT a) (NN division)) c-commands itself path ['PP', 'NP', 'NP', 'NP', 'PRP']
tree 71244 : (NP (NNP North) (NNP American) (NNP Philips) (NNP Corp.)) c-commands itself path ['NP', 'NP', 'PRP']
tree 71244 : (NP
  (NP (DT a) (NN subsidiary))
  (PP
    (IN of)
    (NP
      (NP (NNP N.V) (. .) (NNP Philips))
      (PP (IN of) (NP (DT the) (NNPS Netherlands)))))) c-commands itself path ['NP', 'PRP']
```

Homework 10: Question 2 Review

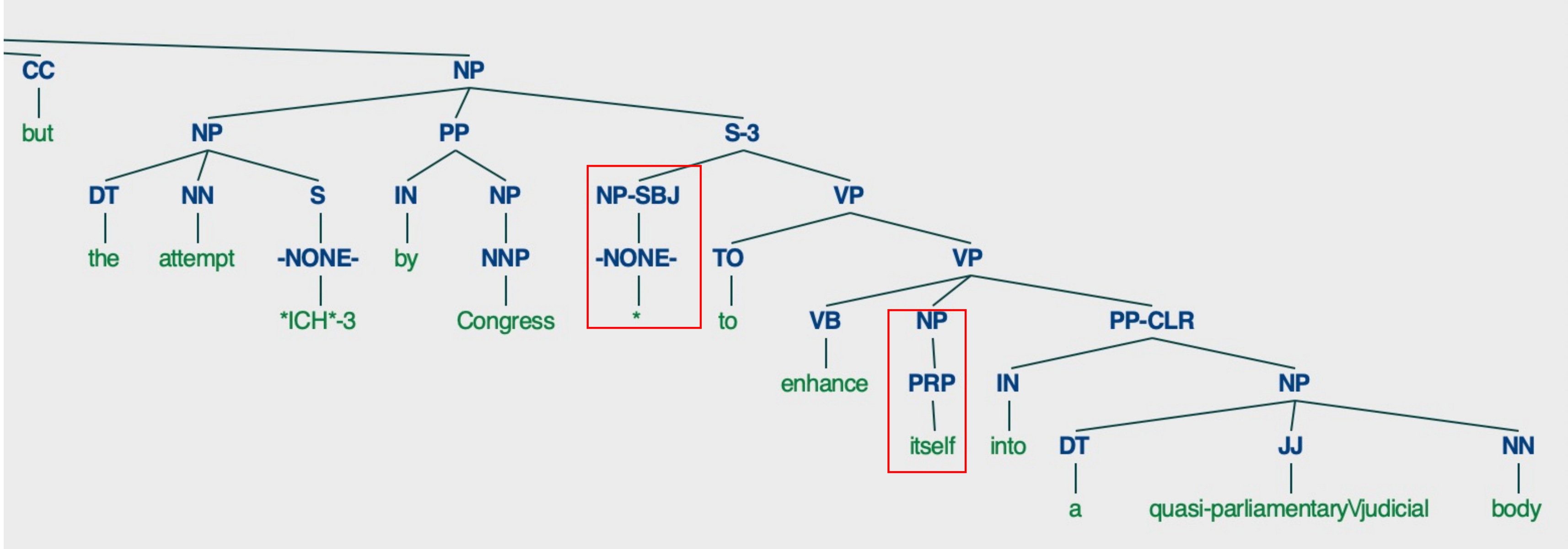


Homework 10: Question 3 Review

- Suppose in the case of multiple candidate NPs, we adopt the rule:
  - the closest NP is the antecedent of the anaphor
- How well does this work in our subcorpus?
- **No need to implement**
  - **Implementation can be tricky:**
    - **given current code,** could report shortest path for anaphor

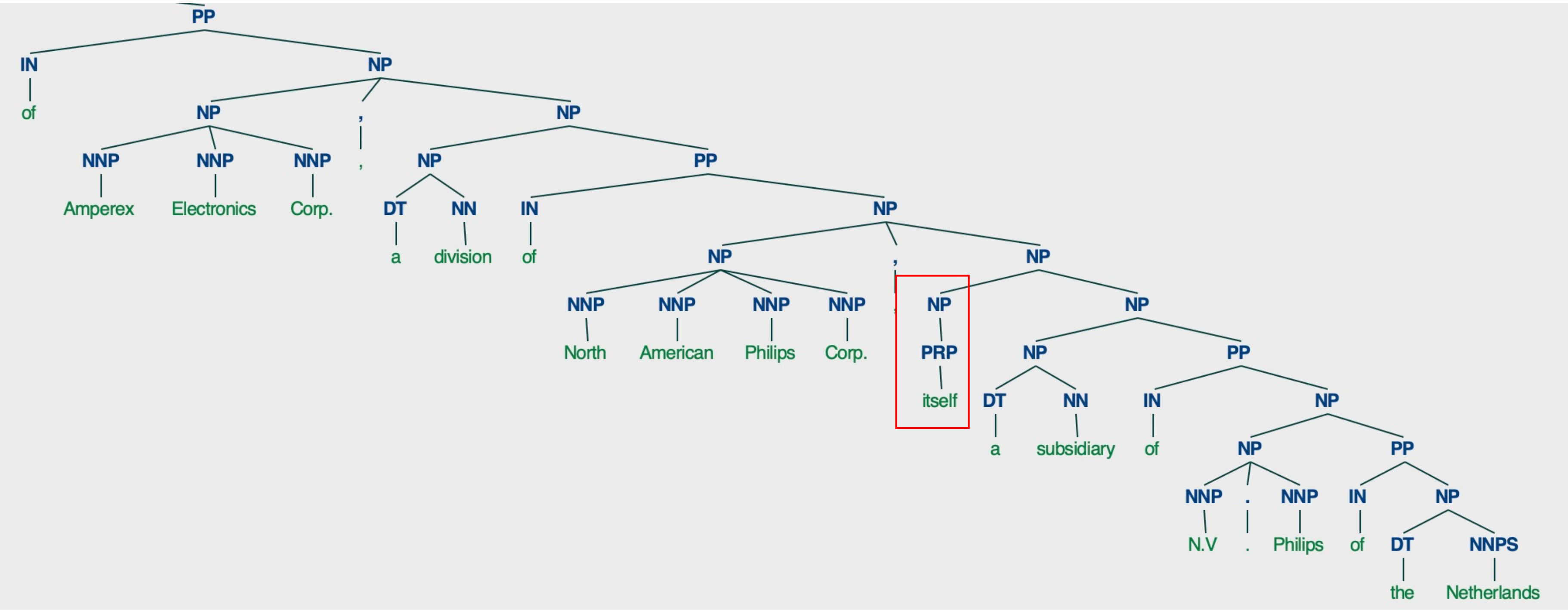
Homework 10: Question 3 Review

- Give both positive and negative examples.



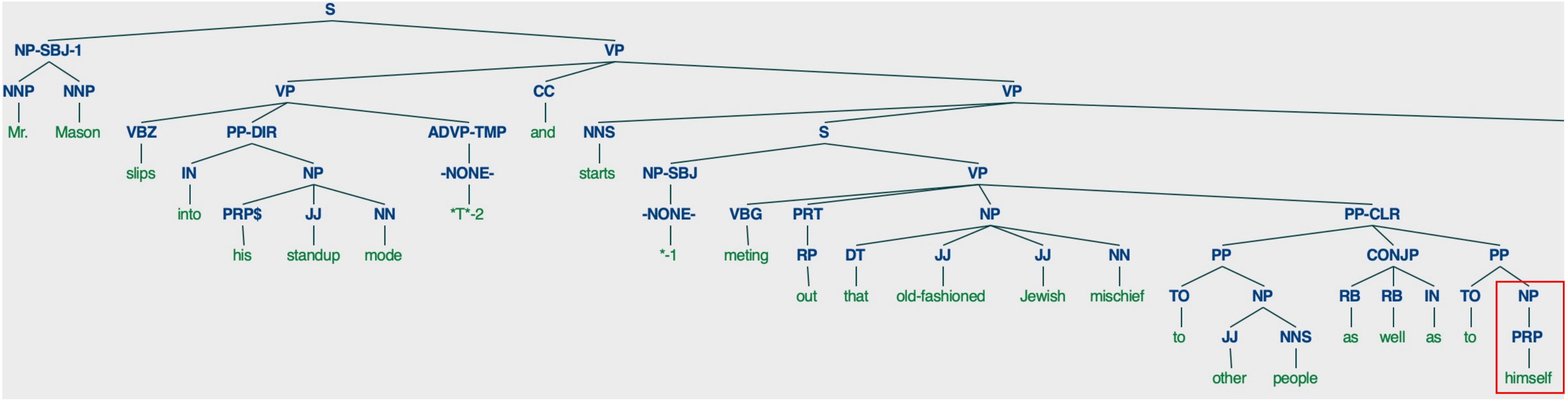
Homework 10: Question 3 Review

- Give both positive and negative examples.



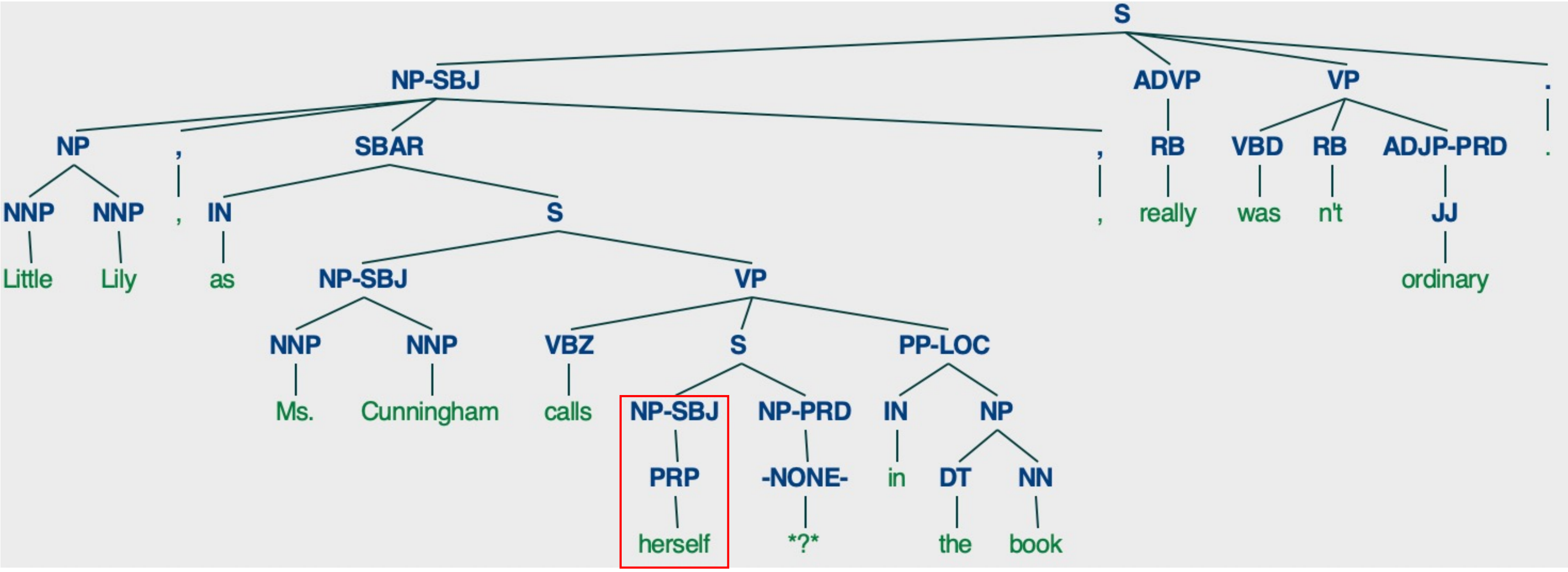
Homework 10: Question 3 Review

- Give both positive and negative examples.



Homework 10: Question 3 Review

- Give both positive and negative examples.

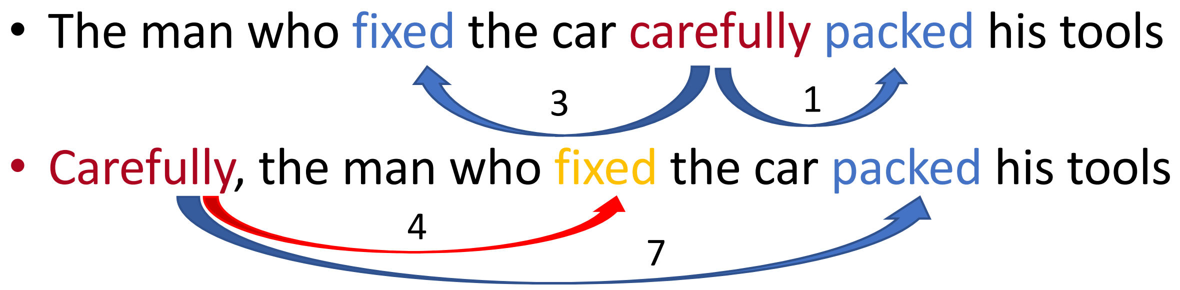


Computing Anaphor Binding

- Actually:
    - `x c−commands y` (and enumerate the possibilities for y) is the wrong definition to use (*as it's top-down*).
    - It should be `y is c−commanded by x` (enumerate the possibilities for x) (*bottom-up*)
- Note that Prof. Chomsky in his talk said:
    - *Well, take another example: Anaphors, terms that lack independent reference have to seek an antecedent, …*

Inadequacies of the purely linear model

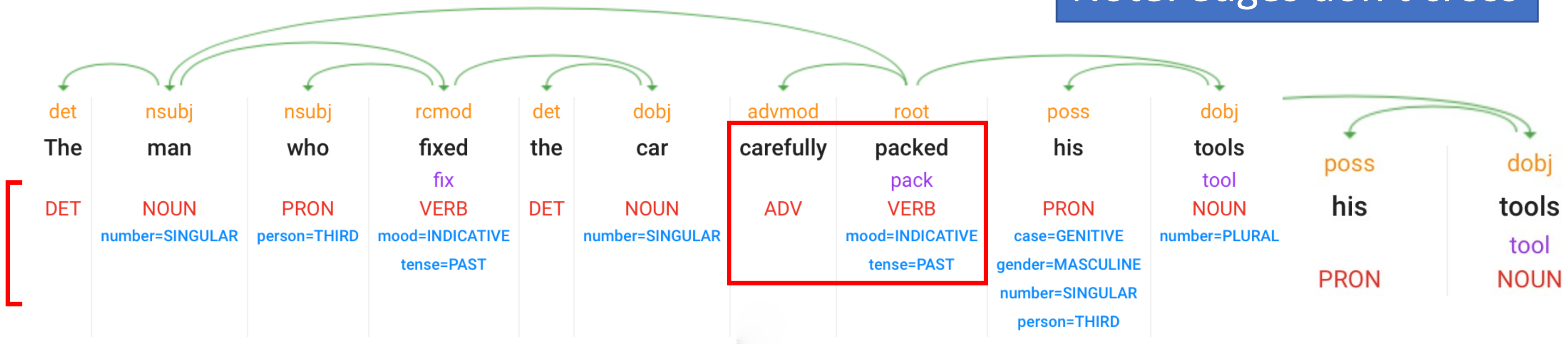
- Examples (Chomsky, *OLLI lecture*, Oct 2021):



  - The man who fixed the car carefully packed his tools
  - Carefully, the man who fixed the car packed his tools

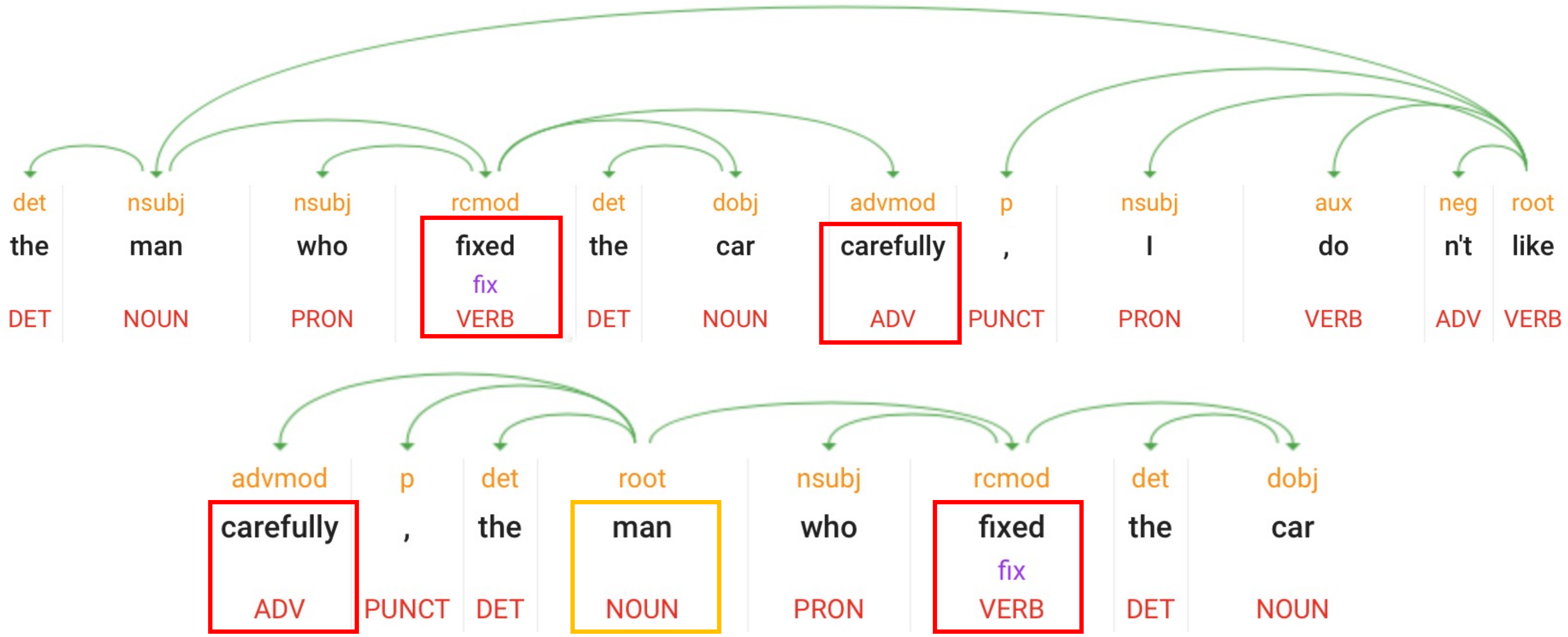
- Google Natural Language parse:

Note: edges don't cross



det	nsubj	nsubj	rcmod	det	dobj	advmod	root	poss	dobj	poss	dobj
The	man	who	fixed	the	car	carefully	packed	his	tools	his	tools
			fix				pack		tool		tool
DET	NOUN	PRON	VERB	DET	NOUN	ADV	VERB	PRON	NOUN	PRON	NOUN
	number=SINGULAR	person=THIRD	mood=INDICATIVE tense=PAST		number=SINGULAR		mood=INDICATIVE tense=PAST	case=GENITIVE gender=MASCULINE number=SINGULAR person=THIRD	number=PLURAL		

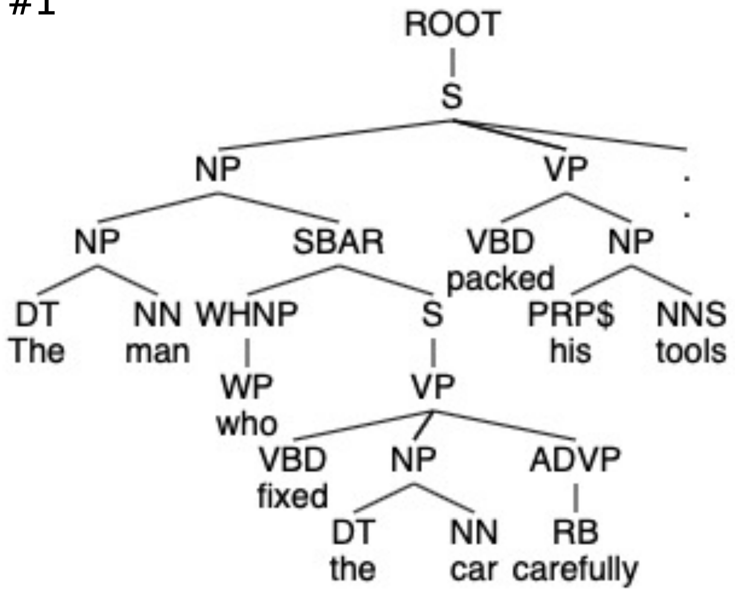
Google Natural Language



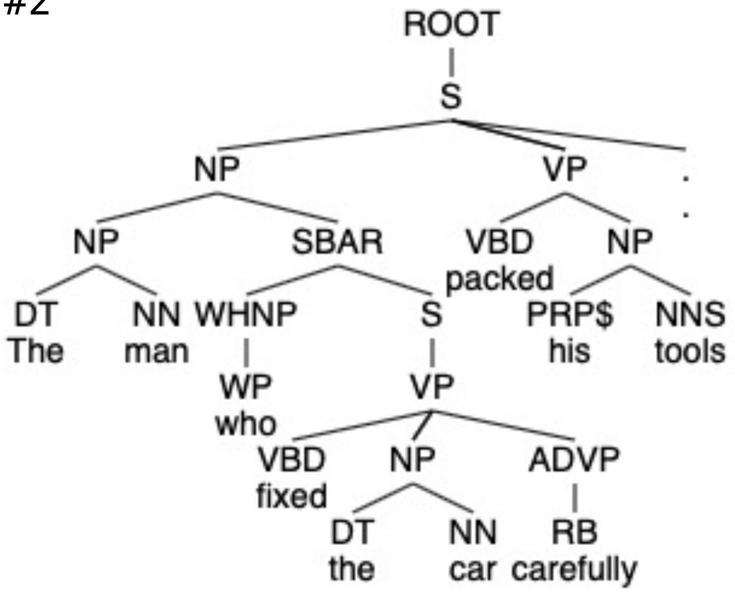
Standalone Stanford Parser

Recall the kbest parser from Homework 7 ? Well, …

#1

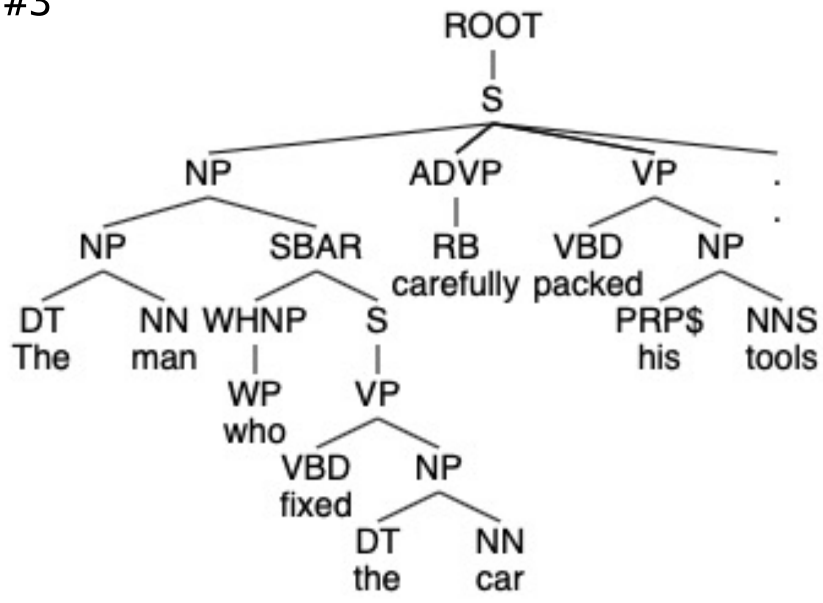


#2

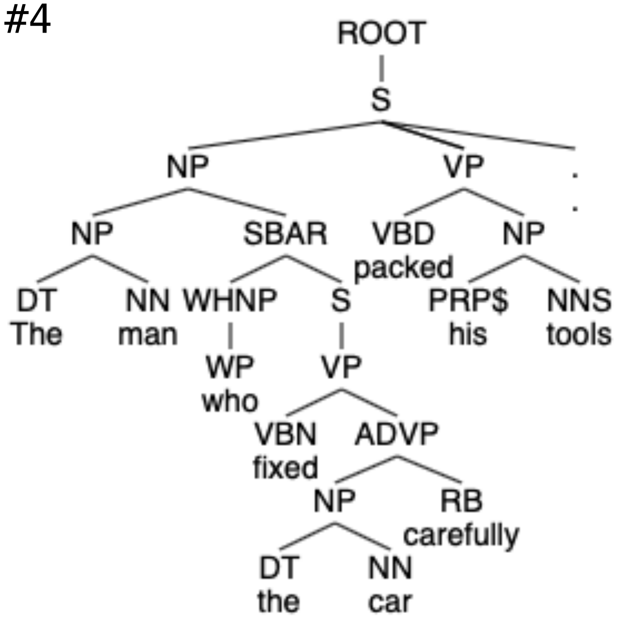


Standalone Stanford Parser

#3

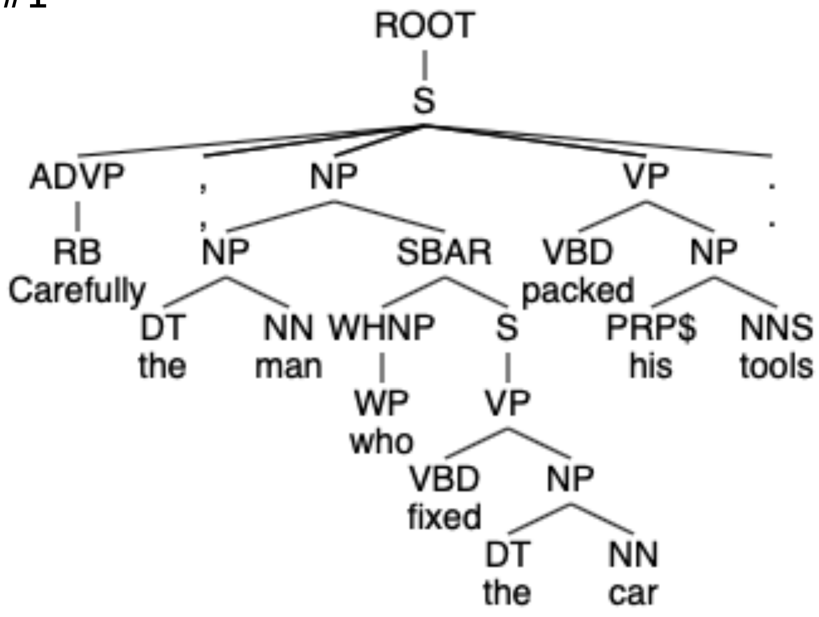


#4

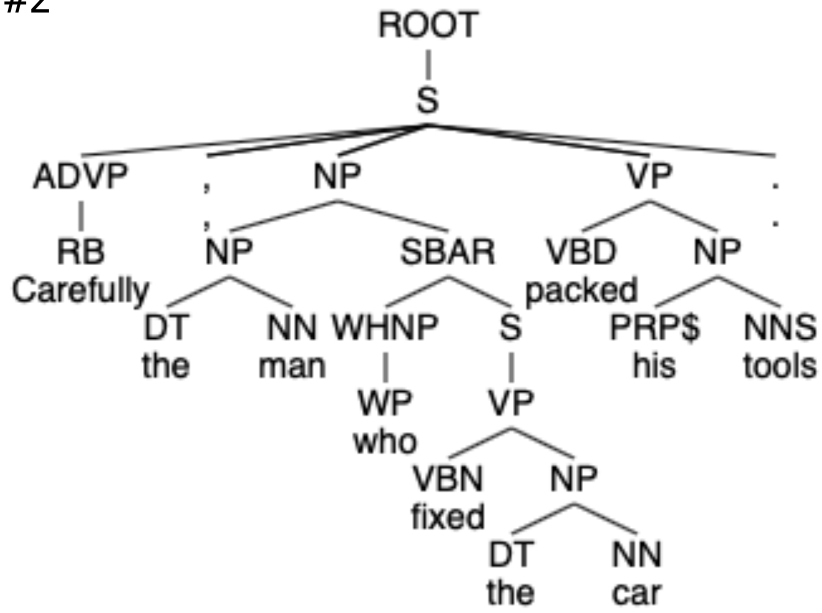


Standalone Stanford Parser

#1



#2


ROOT
S
ADVP
,
NP
VP
.
RB
Carefully
,
NP
SBAR
VBD
packed
NP
.
DT
the
NN
man
WHNP
S
PRP$
his
NNS
tools
WP
who
VP
VBN
fixed
NP
DT
the
NN
car