LING/C SC 581:
Advanced Computational Linguistics

Lecture 18

Today's Topic

- Homework 8 Review
- ptb package in nltk
- c-command

Homework 8: Question 1 Review

Statistics History

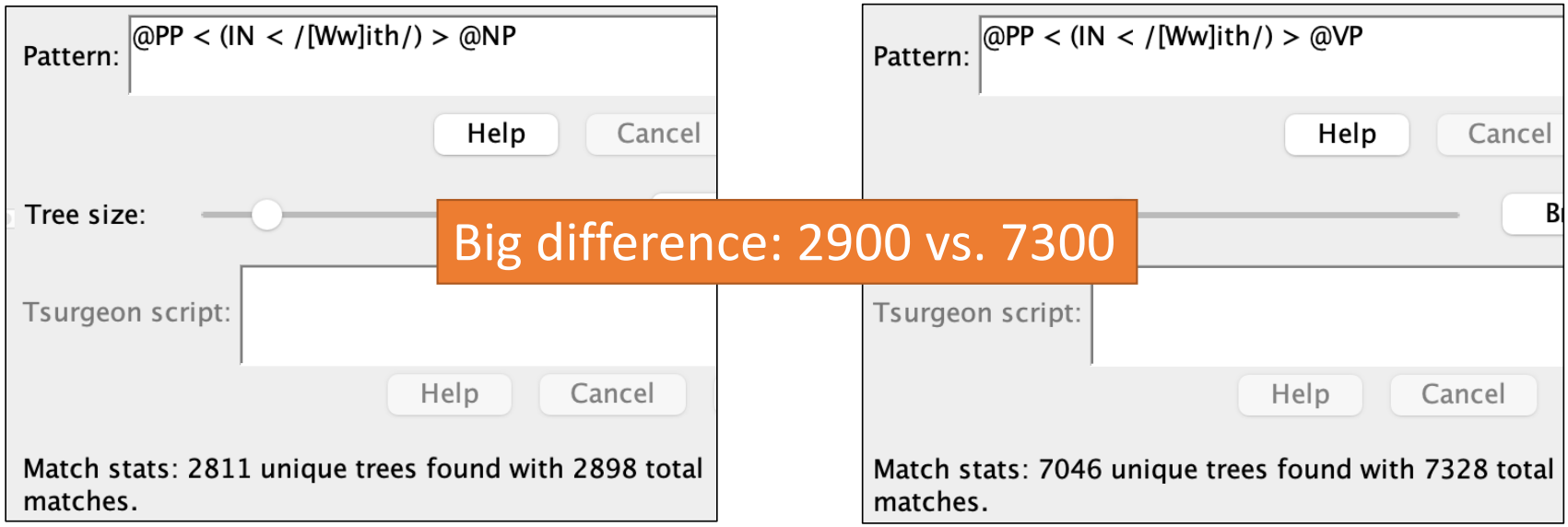
Pattern	Trees Matched	Total Matches
PP > NP	36030	52977
@PP > @NP	52115	86462
PP > VP	31323	38658
@PP > @VP	65417	96161

- Are there more cases of PP attachment to NPs or VPs?
  - 86K vs. 96K (*almost the same*)
  - 53K vs. 39K

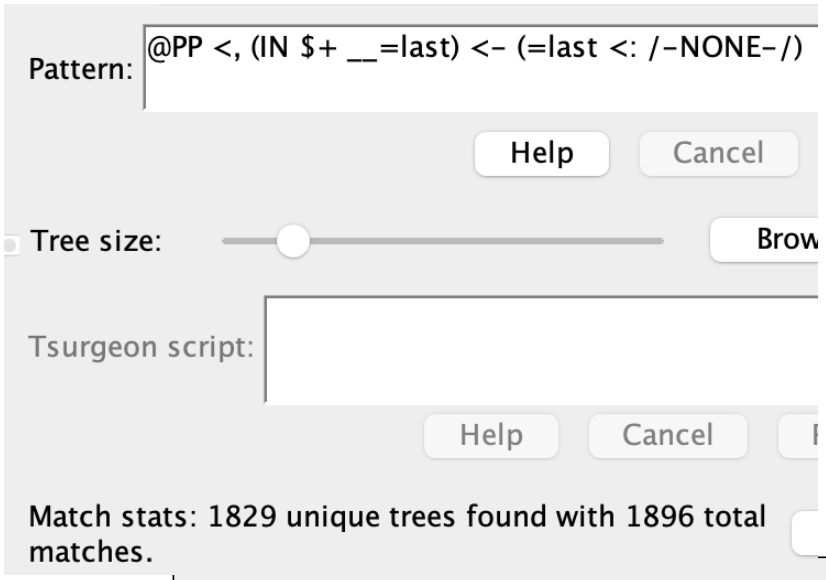
Problem with memory size? Just give java more memory as I explained in class:

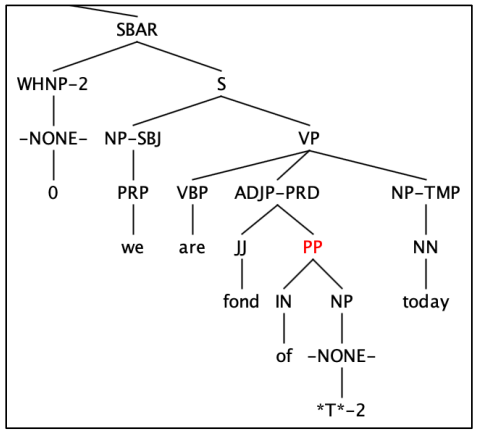
```
2 java -mx2000m -cp `dirname $0`/stanford-tregex.jar edu.stanford.nlp.trees.treg…
  ex.gui.TregexGUI
```

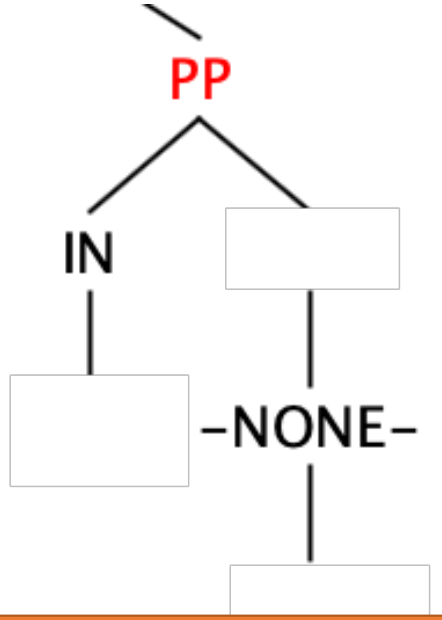
Homework 8: Question 2 Review



Homework 8: Question 3 Review



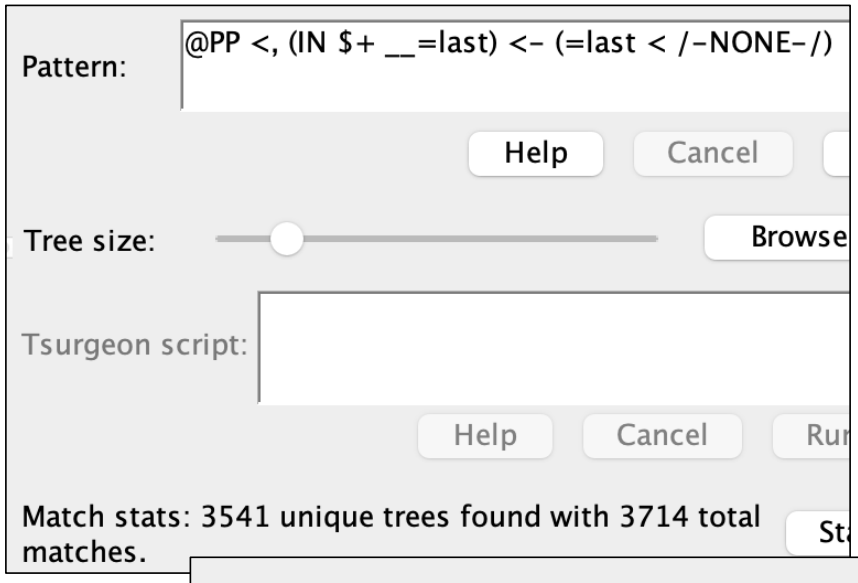


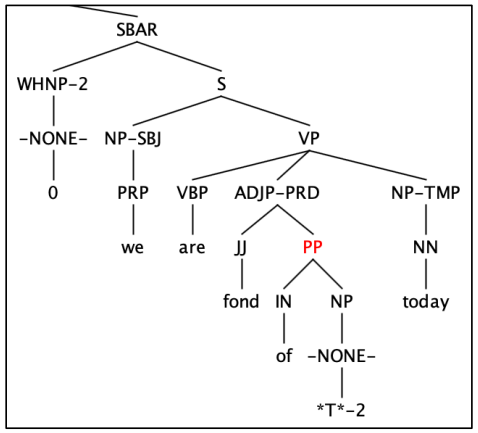


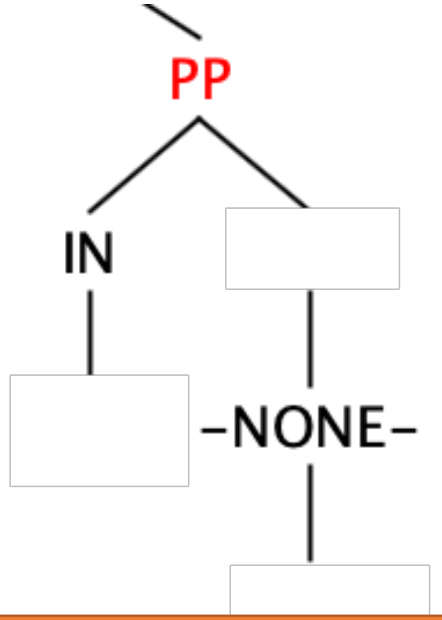
1896 matches in 254K trees (≅ 0.75%)

Browse stats: 253568 trees found in the selected files

Homework 8: Question 3 Review





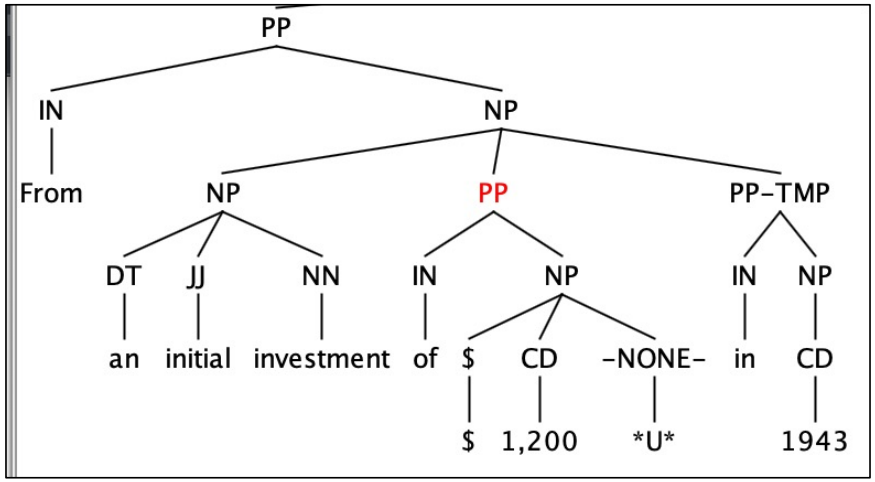


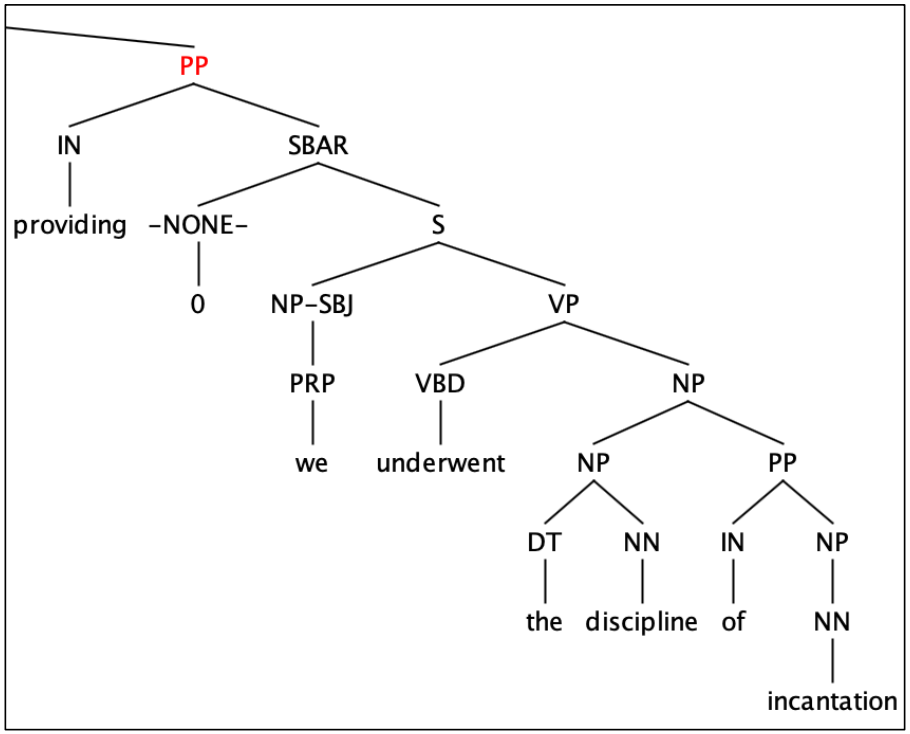
3714 matches in 254K trees (≅ 1.5%)

Browse stats: 253568 trees found in the selected files

Homework 8: Question 3 Review

Overgeneration: < vs <:





Homework 8: Question 3 Review

Corpus	Number	Total	%
Brown	282	24243	1.2
WSJ	246	49208	0.5
SWB	1368	*177804	0.8
ATIS	0	*2309	0

Homework 8: Question 3 Review

swbd

```
 ¶
13¶
14( (CODE (SYM SpeakerA1) (. .) ))¶
15( (INTJ (UH Okay) (. .) (-DFL- E_S) ))¶
16( (S ¶
17    (INTJ (UH Uh) )¶
18    (, ,) ¶
19    (ADVP-TMP (RB first) )¶
20    (, ,) ¶
21    (INTJ (UH um) )¶
22    (, ,) ¶
23    (NP-SBJ-1 (PRP I) )¶
24    (VP (VBP need) ¶
```

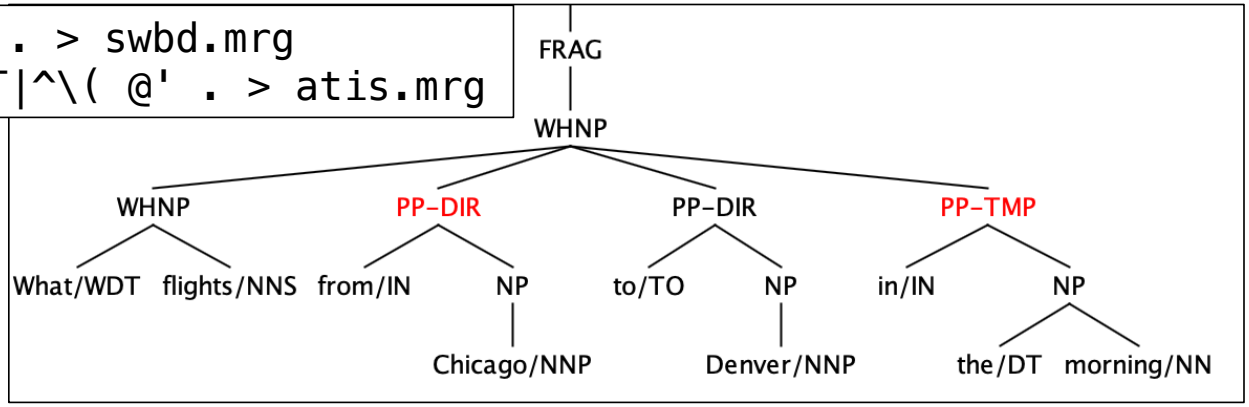
atis

```
 2( @0y0012sx-a-11/CD )¶
 3( END_OF_TEXT_UNIT )¶
 4( (S ¶
 5    (NP-SBJ */XXX )¶
 6    (VP List/VB ¶
 7      (NP ¶
 8        (NP the/DT flights/NNS )¶
 9        (PP-DIR from/IN ¶
10          (NP Baltimore/NNP ))¶
11        (PP-DIR to/TO ¶
12          (NP Seattle/NNP ))¶
```

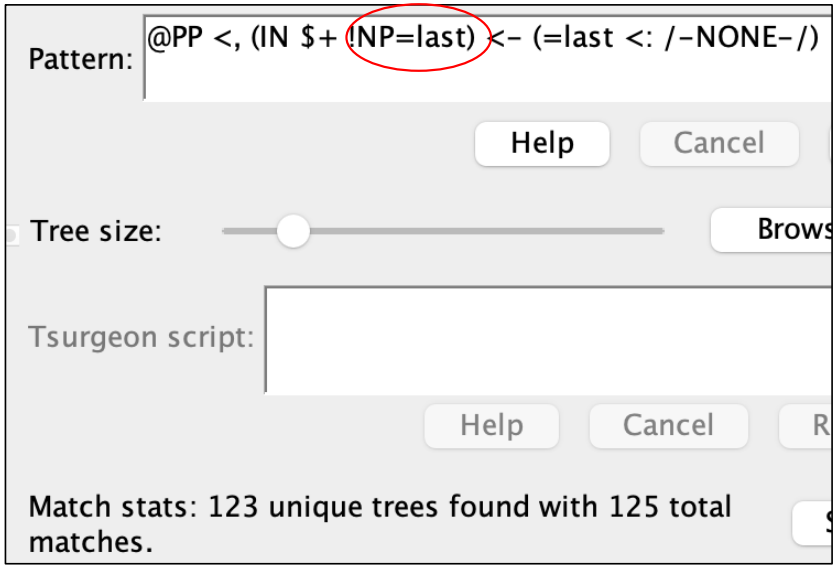
Homework 8: Question 3 Review

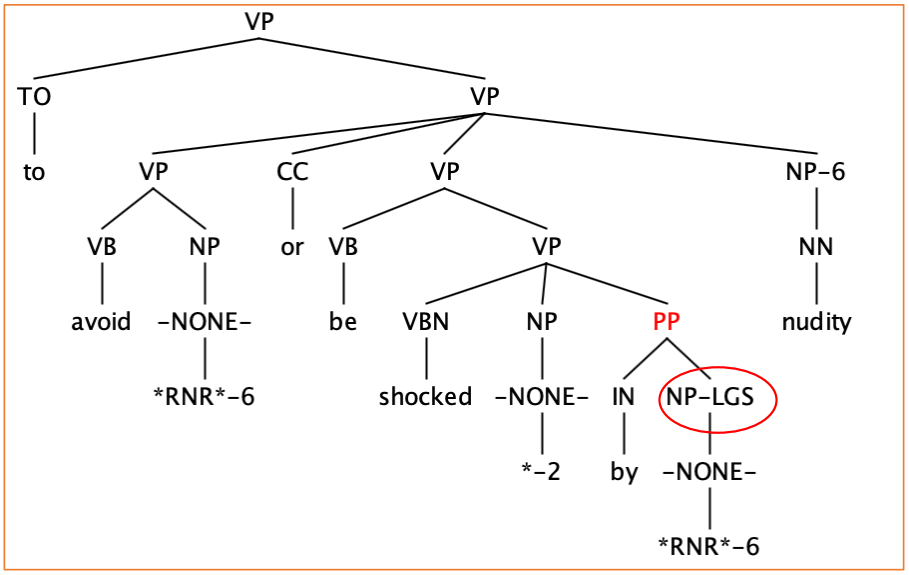
Corpus	Number	Total	%
Brown	282	24243	1.2
WSJ	246	49208	0.5
SWB	1368	111154	1.2
ATIS	0	578	0

```
swbd$ grep -ERvh '\(CODE' . > swbd.mrg
grep -ERvh '\( END_OF_TEXT|^\( @' . > atis.mrg
```



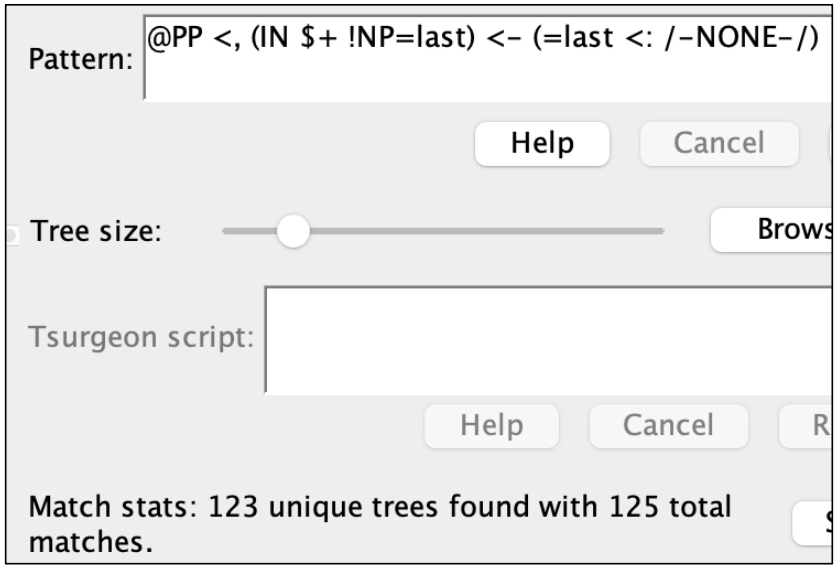
Homework 8: Question 3 Review

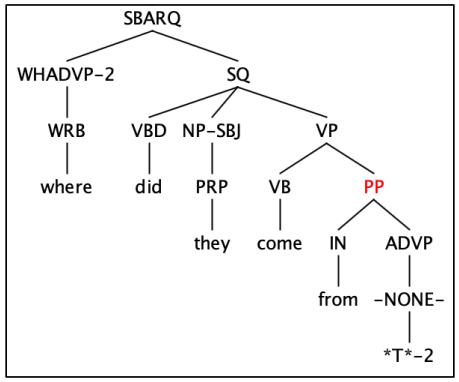




LGS: Logical Subject
RNR: Right Node Raising

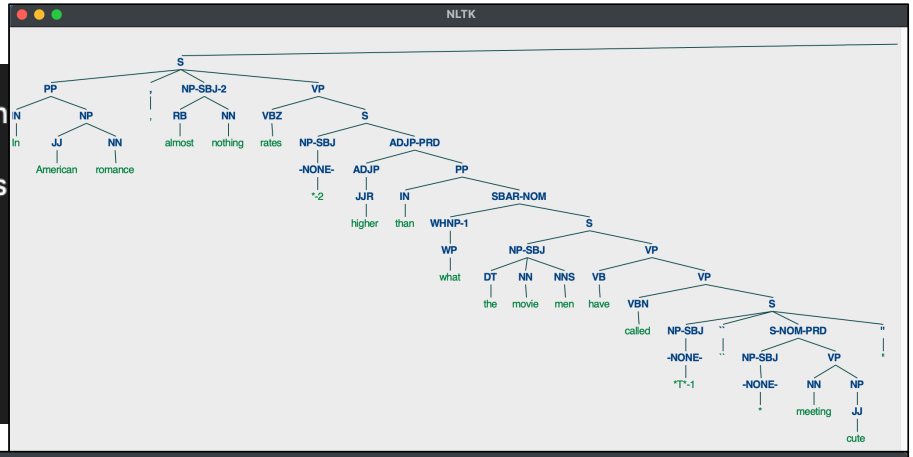
Homework 8: Question 3 Review

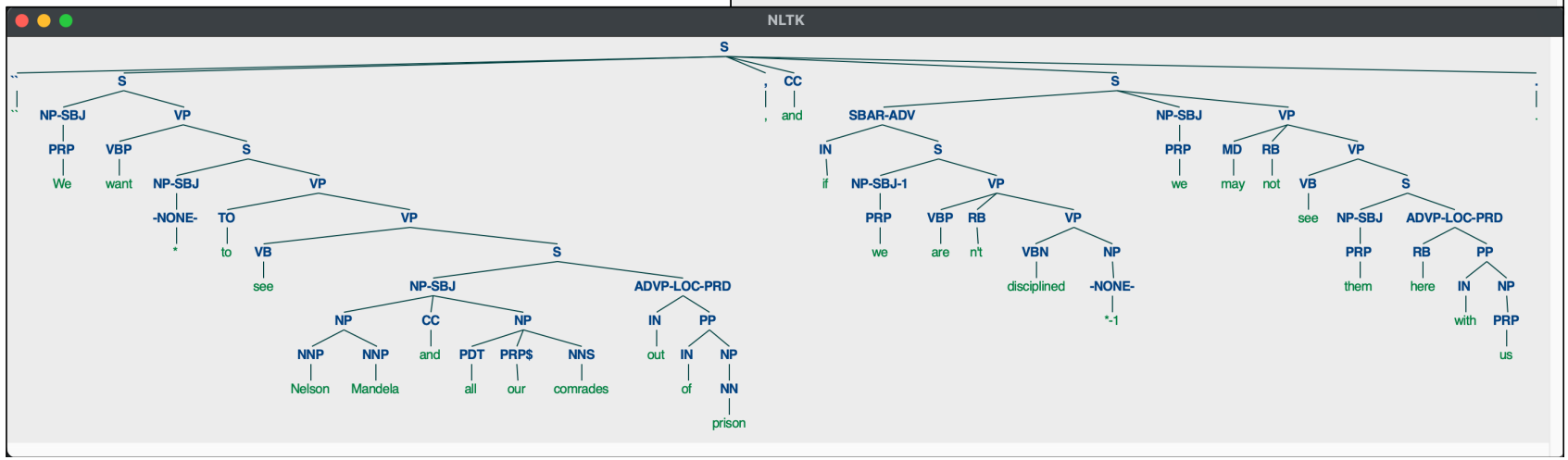




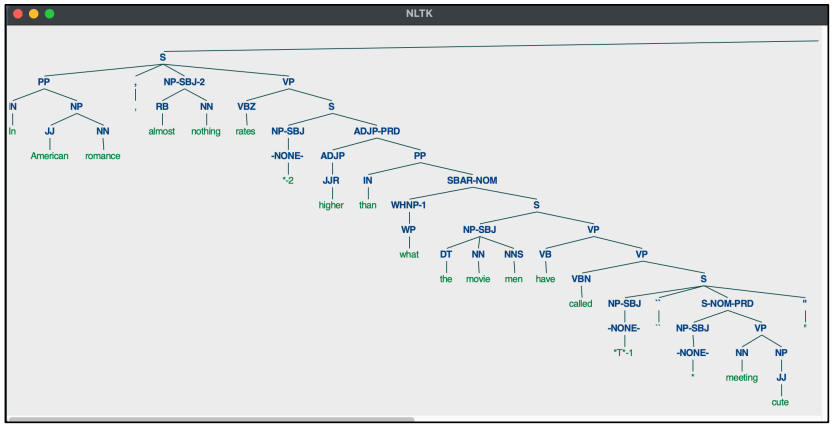
`ptb` package in `nltk`

```
(base) ling581-22$ python
Python 3.9.9 | packaged by conda-forge | (main
[Clang 11.1.0 ] on darwin
Type "help", "copyright", "credits" or "licens
>>> from nltk.corpus import ptb
>>> t1 = ptb.parsed_sents()[0]
>>> tL = ptb.parsed_sents()[-1]
>>> t1.draw()
>>> tL.draw()
>>>
```





`ptb` package in `nltk`



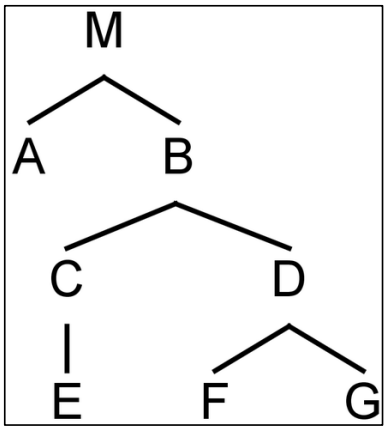

```
>>> t1.label()
'S'
>>> t1.height()
16
>>> t1[0]
Tree('S', [Tree('PP', [Tree('IN', ['In']), Tree('NP', [Tree('JJ', ['American']), Tree('NN', ['romance'])])]), Tree(',', [',']), Tree('NP-SBJ-2', [Tree('RB', ['almost']), Tree('NN', ['nothing'])]),
Tree('VP', [Tree('VBZ', ['rates']), Tree('S', [Tree('NP-SBJ', [Tree('-NONE-', ['*-2'])]), Tree('ADJP-PRD', [Tree('ADJP', [Tree('JJR', ['higher']), Tree('PP', [Tree('IN', ['than']), Tree('SBAR-
NOM', [Tree('WHNP-1', [Tree('WP', ['what'])]), Tree('S', [Tree('NP-SBJ', [Tree('DT', ['the']), Tree('NN', ['movie']), Tree('NNS', ['men'])]), Tree('VP', [Tree('VB', ['have']), Tree('VP',
[Tree('VBN', ['called']), Tree('S', [Tree('NP-SBJ', [Tree('-NONE-', ['*T*-1'])]), Tree('``', ['``']), Tree('S-NOM-PRD', [Tree('NP-SBJ', [Tree('-NONE-', ['*'])]), Tree('VP', [Tree('NN', ['meeting']),
Tree('NP', [Tree('JJ', ['cute'])])])]), Tree("''", ["''"])])])])])])])])])])])

>>> t1[1]
Tree(':', ['--'])
>>> t1[2]
Tree('S', [Tree('S-ADV', [Tree('NP-SBJ', [Tree('DT', ['that'])]), Tree('VP', [Tree('VBZ', ['is'])])]), Tree(',', [',']), Tree('NP-SBJ', [Tree('NN', ['boy-meets-girl'])]), Tree('VP', [Tree('VBZ',
['seems']), Tree('ADJP-PRD', [Tree('RB', ['more']), Tree('JJ', ['adorable'])]), Tree('SBAR-ADV', [Tree('IN', ['if']), Tree('S', [Tree('NP-SBJ', [Tree('PRP', ['it'])]), Tree('VP', [Tree('VBZ',
['does']), Tree('RB', ["n't"]), Tree('VP', [Tree('VB', ['take']), Tree('NP', [Tree('NN', ['place'])]), Tree('PP', [Tree('IN', ['in']), Tree('NP', [Tree('NP', [Tree('DT', ['an']), Tree('NN',
['atmosphere'])]), Tree('PP', [Tree('IN', ['of']), Tree('NP', [Tree('ADJP', [Tree('JJ', ['correct']), Tree('CC', ['and']), Tree('JJ', ['acute'])]), Tree('NN', ['boredom'])])])])])])])])])])])])

>>> t1[3]
Tree('.', ['.'])
>>> t1[4]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/opt/miniconda3/lib/python3.9/site-packages/nltk/tree/tree.py", line 156, in __getitem__
    return list.__getitem__(self, index)
IndexError: list index out of range
>>> len(t1)
4
```

nltk.tree.tree module
https://www.nltk.org/api/nltk.tree.tree.html

`ptb` package in `nltk`

- Linguistic definition (logic):
  - X **c-commands** Y iff ∃ Z, W such that Z < X and Z < W, W ≠ X, (W = Y or W << Y).

https://en.wikipedia.org/wiki/C-command



- M **does not** c-command any node because it dominates all other nodes.
- A c-commands B, C, D, E, F, and G.
- B c-commands A.
- C c-commands D, F, and G.
- D c-commands C and E.
- E **does not** c-command any node because it does not have a sister node or any daughter nodes.
- F c-commands G.
- G c-commands F.

`ptb` package in `nltk`

```python
# (c) Sandiway Fong, University of Arizona, 2022
from itertools import permutations
from nltk.tree import Tree
t1 = Tree.fromstring("(S (NP I) (VP (V saw) (NP him)))")
t2 = Tree.fromstring("(M A (B (C E) (D F G)))")

def dom(x):
    yield x
    if not isinstance(x, str):
        for y in x:
            yield from dom(y)

def cc(x):
    if len(x) > 1:
        for y,z in permutations(x, 2):
            for w in dom(z):
                print(y, 'c-commands', w)
        for u in x:
            cc(u)
```

- `ccommand.py`

`itertools.`**`permutations`**`(`*`iterable`*`, `*`r=None`*`)`

Return successive *r* length permutations of elements in the *iterable*.

Why `yield from`?

A Python generator is a form of coroutine, but has the limitation that it can only yield to its immediate caller. This means that a piece of code containing a `yield` cannot be factored out and put into a separate function in the same way as other code. Performing such a factoring causes the called function to itself become a generator, and it is necessary to explicitly iterate over this second generator and re-yield any values that it produces.

`ptb` package in `nltk`

- Let's test the code as follows:
  - `python –i ccommand.py`
  - `cc(t1)`

`ptb` package in `nltk`

```
# (c) Sandiway Fong, University of Arizona, 2022
from itertools import permutations
from nltk.tree import Tree
t1 = Tree.fromstring("(S (NP I) (VP (V saw) (NP him)))")
t2 = Tree.fromstring("(M A (B (C E) (D F G)))")
t3 = Tree.fromstring("(TOP (S (NP I) (VP (V saw) (NP him))))")

def dom(x):
    yield x
    if not isinstance(x, str):
        for y in x:
            yield from dom(y)

def cc(x):
    if not isinstance(x, str):
        if len(x) > 1:
            for y,z in permutations(x, 2):
                for w in dom(z):
                    print(y, 'c-commands', w)
            for u in x:
                cc(u)
        else:
            cc(x[0])
```

- `ccommand2.py`
- Handles unary branching:
  - e.g. `(TOP (S … ))` in t3