Name: Syed Ahtsham Ul Hassan

Course: Advanced Natural Language Processing

Assignment: 04 – CKY Parsing

Date: 15 January, 2024

CKY Parsing – ATIS Grammar

Tools Used:

1. IDE: PyCharm
2. Python: NLTK, TreeWidget, CanvasFrame

Motivation of the ATIS Grammar:

The ATIS (Airline Travel Information System) grammar is motivated by the need for spoken dialog systems in air travel. It is designed to facilitate natural language interactions related to airline information. The grammar is more practically motivated by concerns related to providing accurate and useful information in the context of air travel conversations.

Analysis of Sentences with Structural Ambiguity with Syntactic Trees:

Sentence 1:

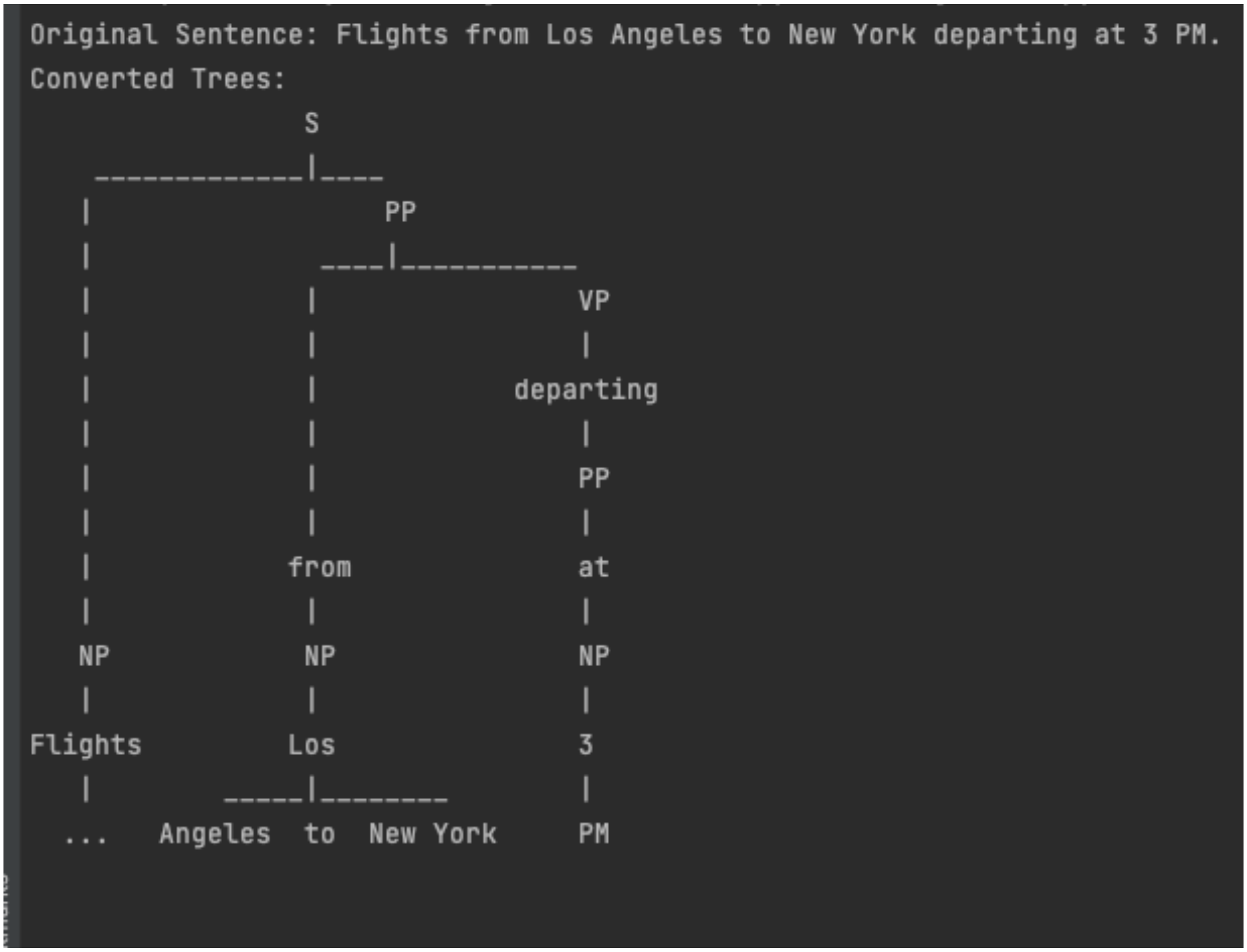
"Flights from Los Angeles to New York departing at 3 PM."

Ambiguity:

The prepositional phrase "from Los Angeles to New York departing at 3 PM" can be interpreted as a modifier of the origin and destination of flights or as a modifier of the time of departure.

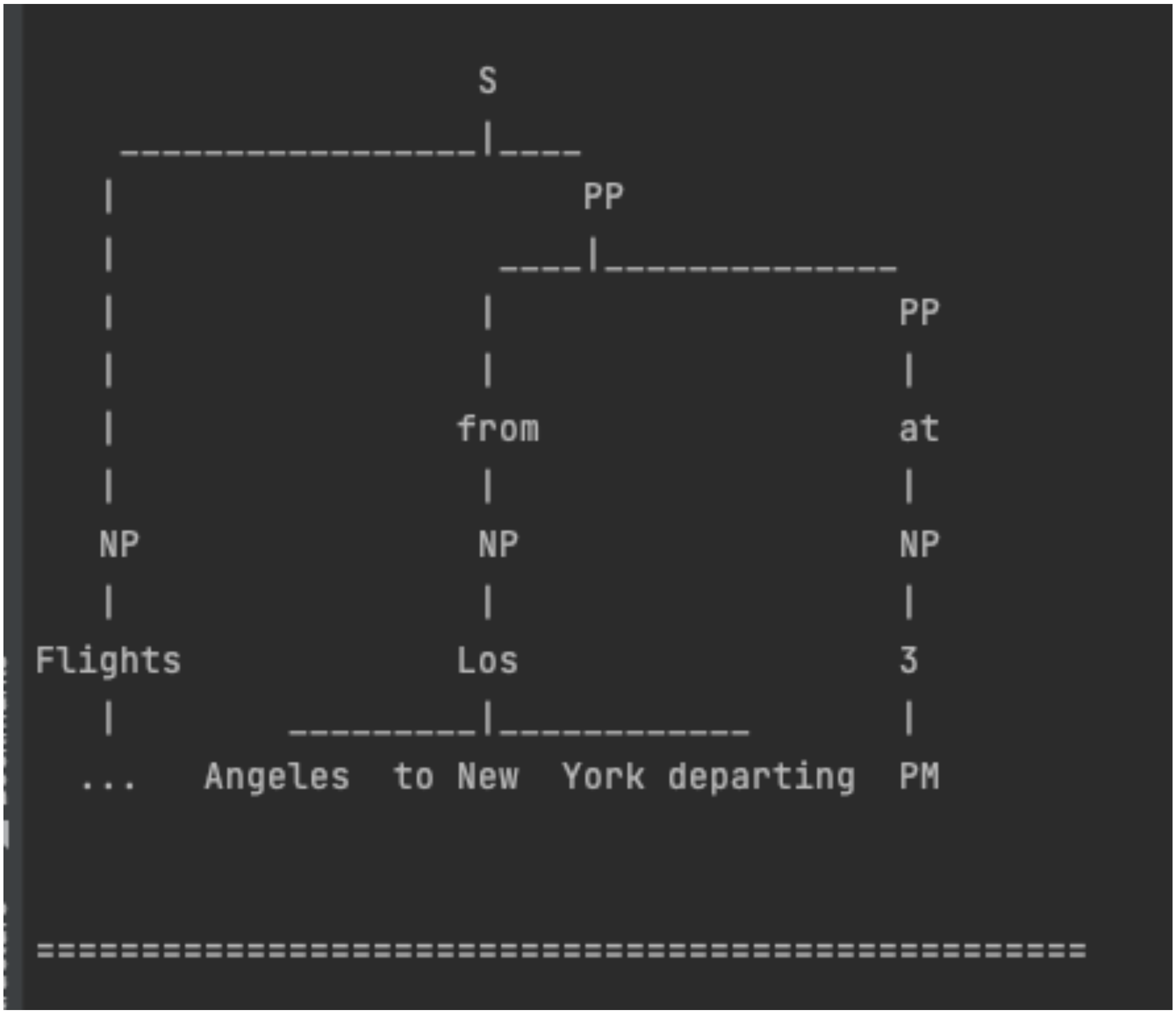
Syntactic Trees:

Analysis 1: (S (NP (Flights)) (PP (from (NP (Los Angeles to New York))) (VP (departing (PP (at (NP (3 PM))))))))



```
Original Sentence: Flights from Los Angeles to New York departing at 3 PM.
Converted Trees:
                 S
    _____________|____
   |                  PP
   |              ____|___________
   |             |                VP
   |             |                |
   |             |            departing
   |             |                |
   |             |                PP
   |             |                |
   |            from              at
   |             |                |
   NP            NP               NP
   |             |                |
Flights         Los               3
   |        _____|________        |
  ...   Angeles  to  New York     PM
```

Analysis 2: (S (NP (Flights)) (PP (from (NP (Los Angeles to New York departing))) (PP (at (NP (3 PM))))))



```
                     S
    _________________|____
   |                      PP
   |                  ____|______________
   |                 |                   PP
   |                 |                   |
   |                from                 at
   |                 |                   |
   NP                NP                  NP
   |                 |                   |
Flights             Los                  3
   |        _________|____________       |
  ...   Angeles  to New  York departing  PM


==================================================
```

Sentence 2:

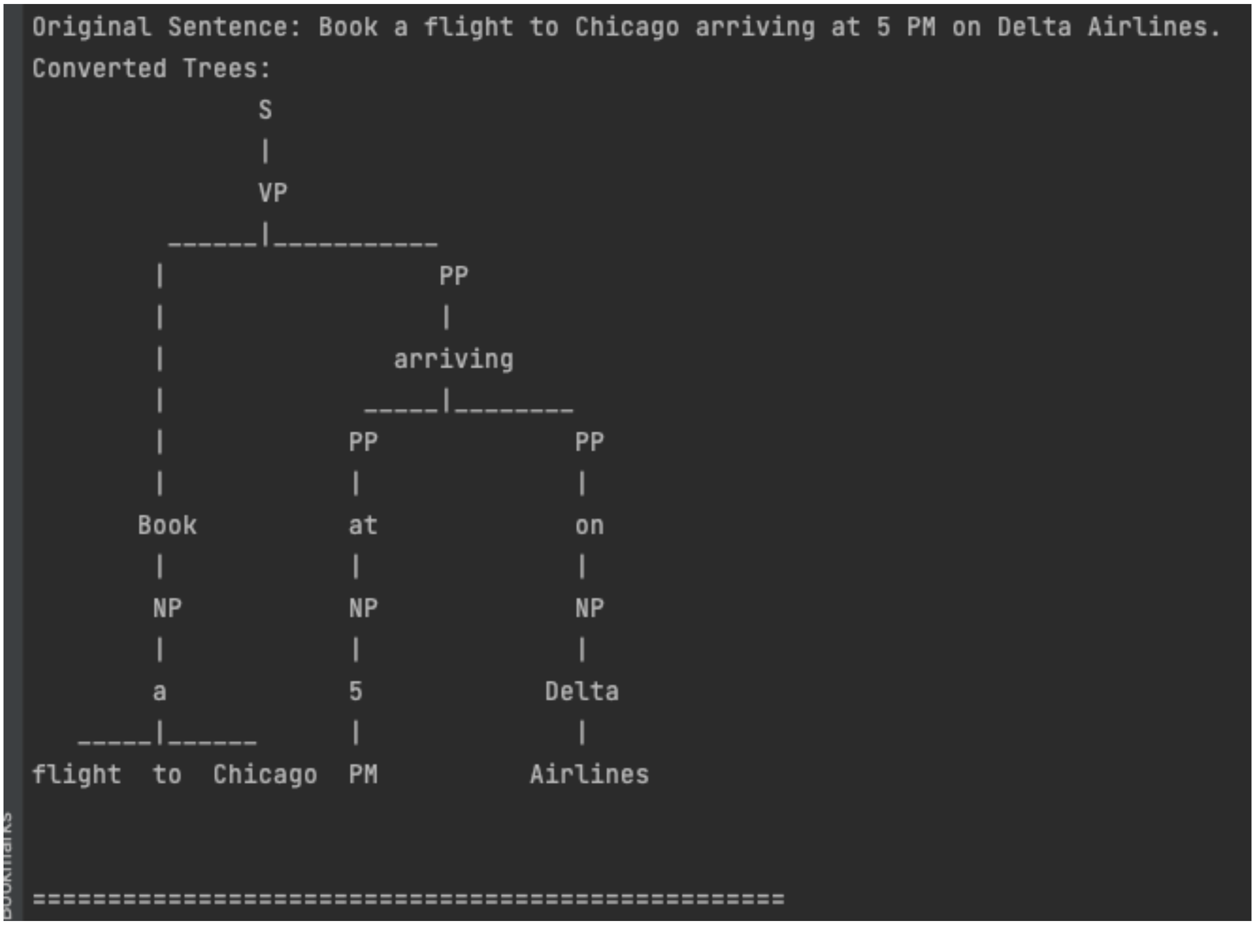
"Book a flight to Chicago arriving at 5 PM on Delta Airlines."

Ambiguity:

The prepositional phrase "to Chicago arriving at 5 PM on Delta Airlines" can be interpreted as a modifier of the destination and arrival time or as a modifier of the airline information.

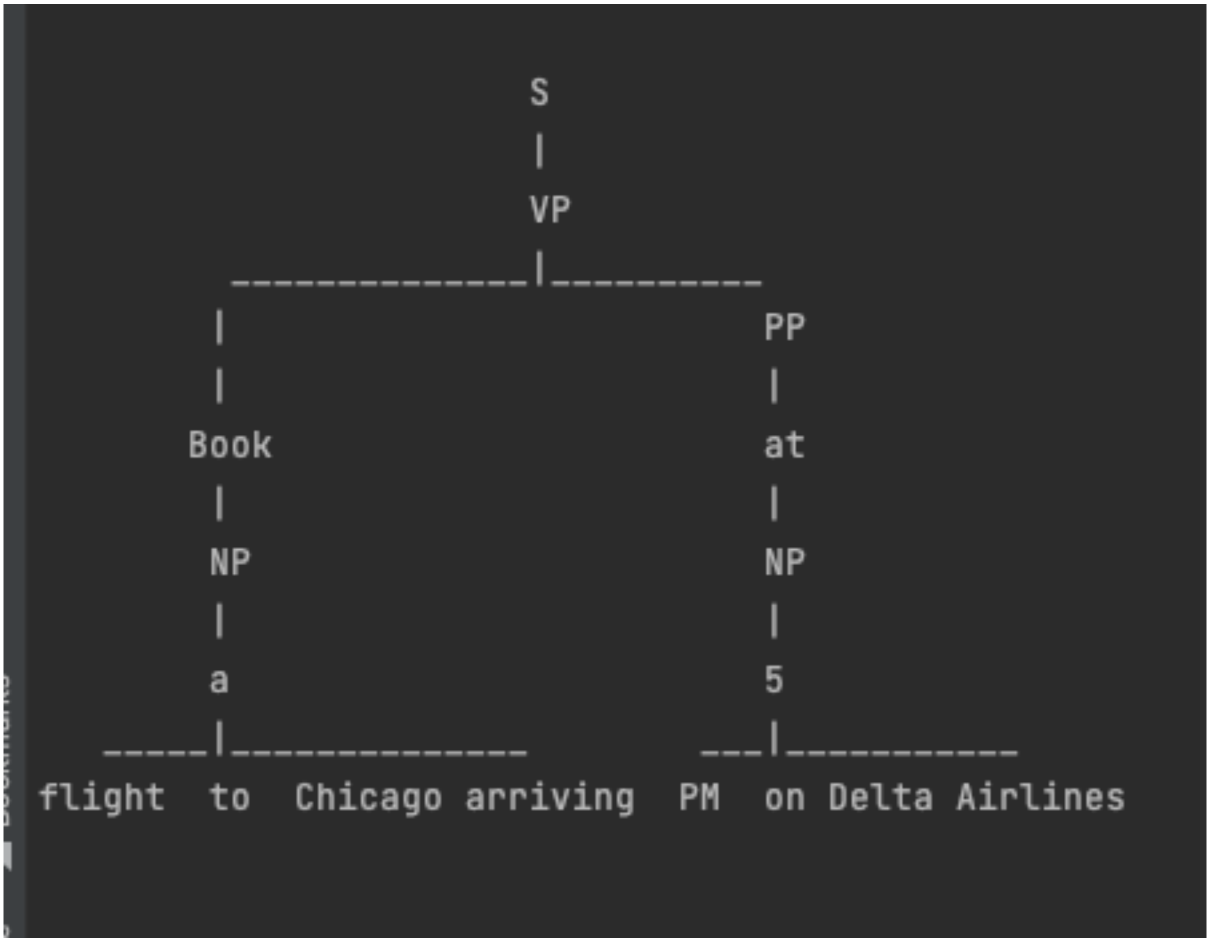
Syntactic Trees:

Analysis 1: (S (VP (Book (NP (a flight to Chicago))) (PP (arriving (PP (at (NP (5 PM)))) (PP (on (NP (Delta Airlines))))))))



```
Original Sentence: Book a flight to Chicago arriving at 5 PM on Delta Airlines.
Converted Trees:
               S
               |
               VP
         ______|___________
        |                  PP
        |                  |
        |               arriving
        |             _____|________
        |            PP             PP
        |            |              |
       Book         at             on
        |            |              |
        NP           NP             NP
        |            |              |
        a            5            Delta
   _____|______      |              |
flight  to  Chicago  PM          Airlines



==================================================
```

Analysis 2: (S (VP (Book (NP (a flight to Chicago arriving))) (PP (at (NP (5 PM on Delta Airlines))))))



```
                       S
                       |
                       VP
         ______________|__________
        |                         PP
        |                         |
       Book                       at
        |                         |
        NP                        NP
        |                         |
        a                         5
   _____|______________        ___|___________
flight  to  Chicago arriving  PM  on Delta Airlines
```

In both sentences, the structural ambiguity arises from different possible interpretations of prepositional phrases, leading to multiple valid syntactic structures.

Discussing Structural Differences:

Number of Parse Trees (Structural Differences):

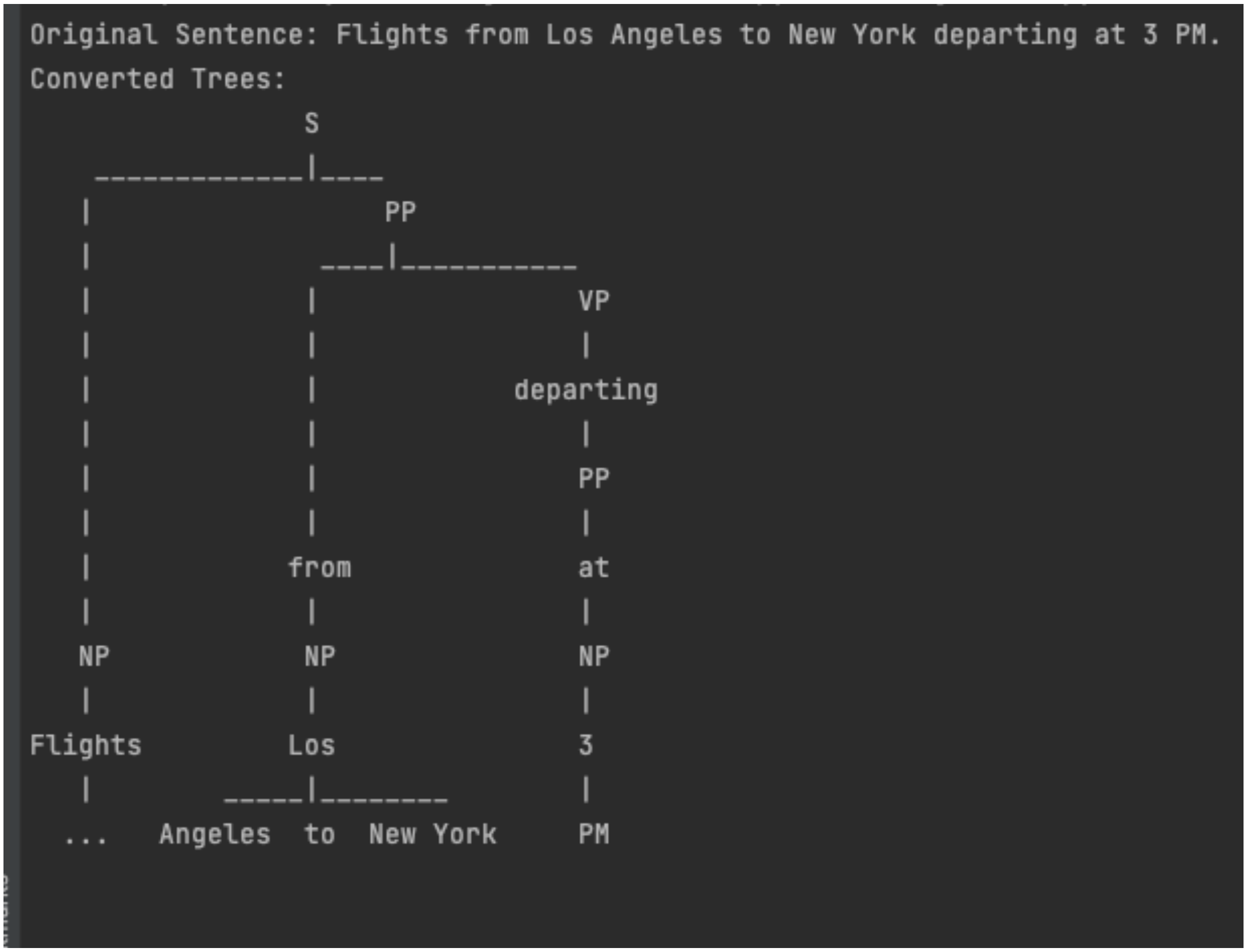
For a given sentence, the CKY parser may generate multiple valid parse trees. The structural differences arise from the different ways the sentence can be syntactically analyzed.

Example with p Parses (p > 1):

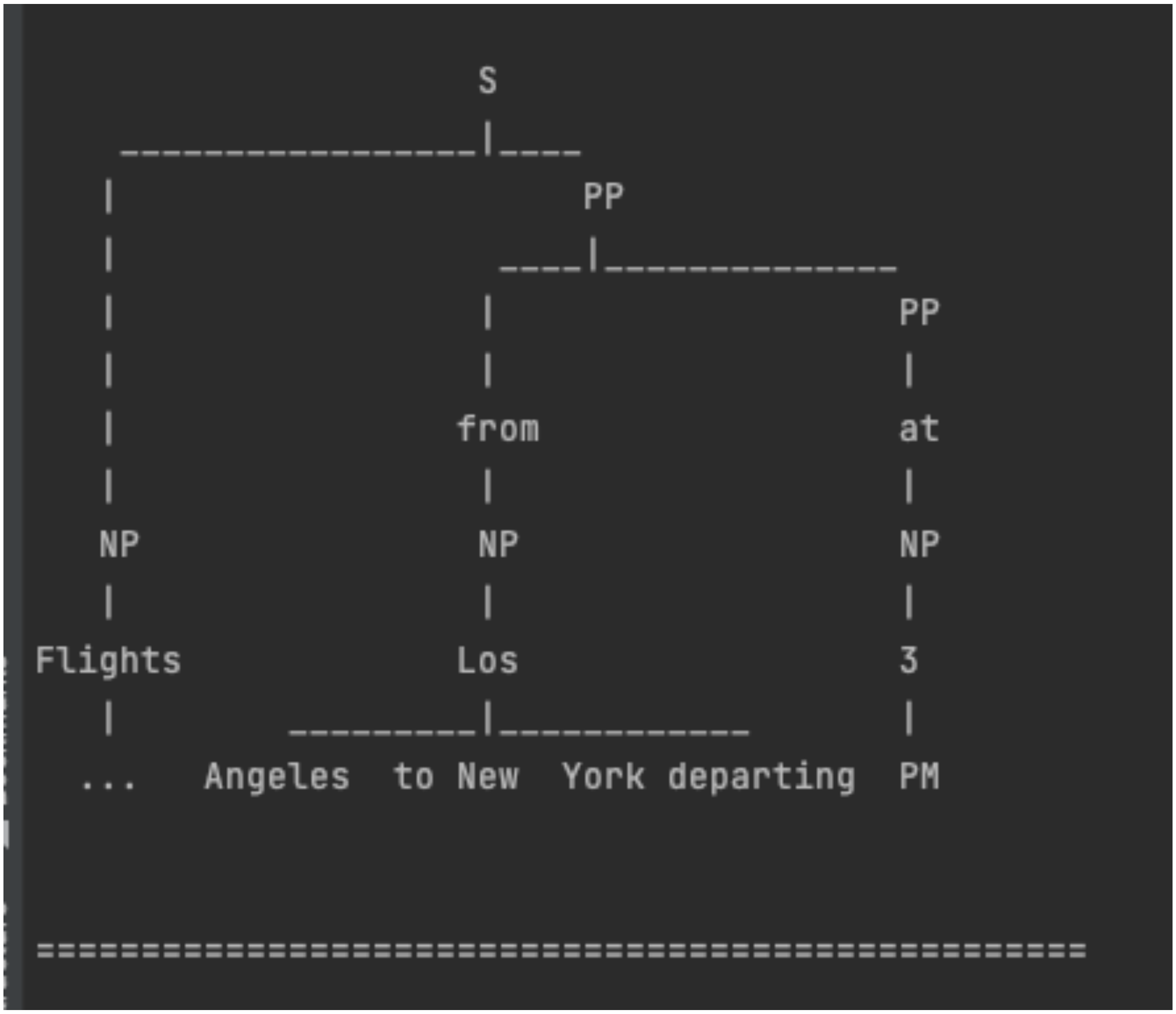
Let's consider an example sentence with p parses such that 1 < p < 5.

Sentence: "Flights from Los Angeles to New York departing at 3 PM."

Structural Differences: Different parses may involve variations in the attachment of prepositional phrases, verb phrases, or other syntactic structures.

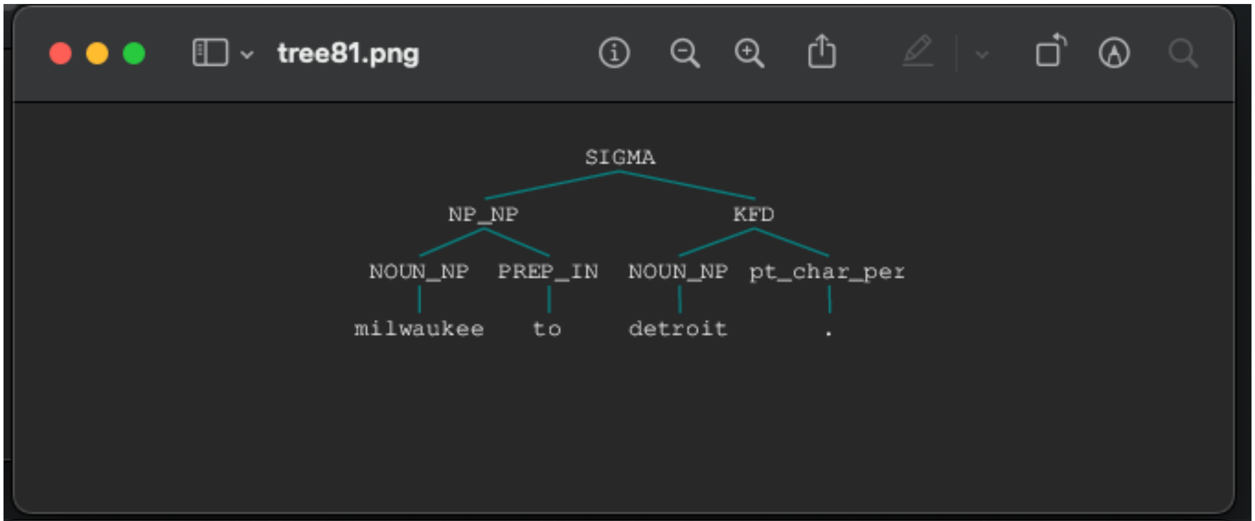


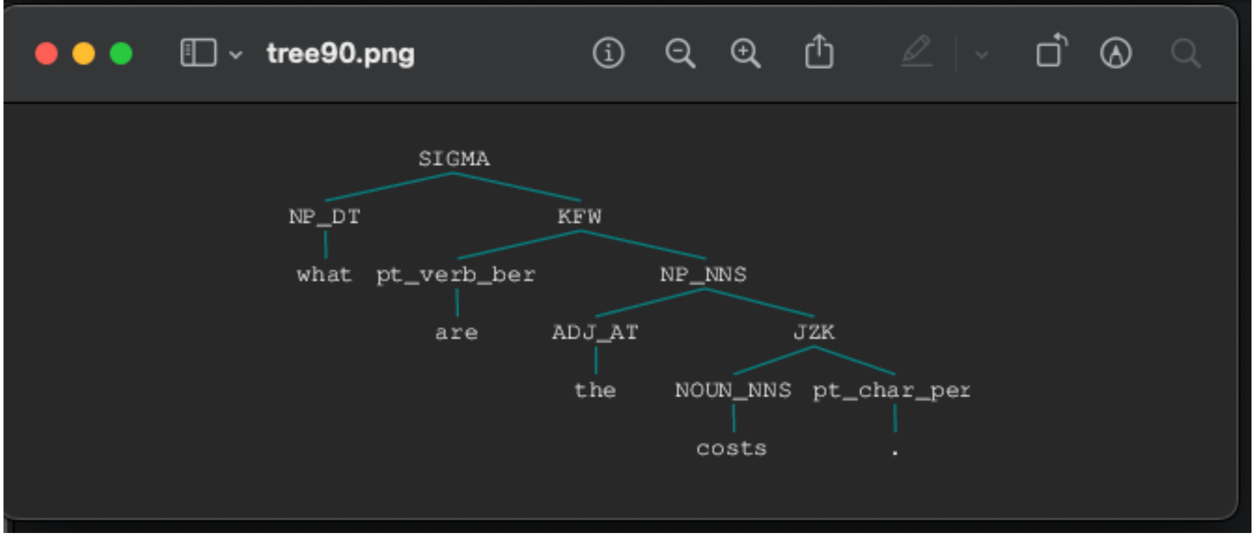
```
Original Sentence: Flights from Los Angeles to New York departing at 3 PM.
Converted Trees:
                 S
    _____________|____
   |                  PP
   |              ____|___________
   |             |                VP
   |             |                |
   |             |            departing
   |             |                |
   |             |                PP
   |             |                |
   |            from              at
   |             |                |
   NP            NP               NP
   |             |                |
Flights         Los               3
   |        _____|________        |
  ...   Angeles  to  New York     PM
```

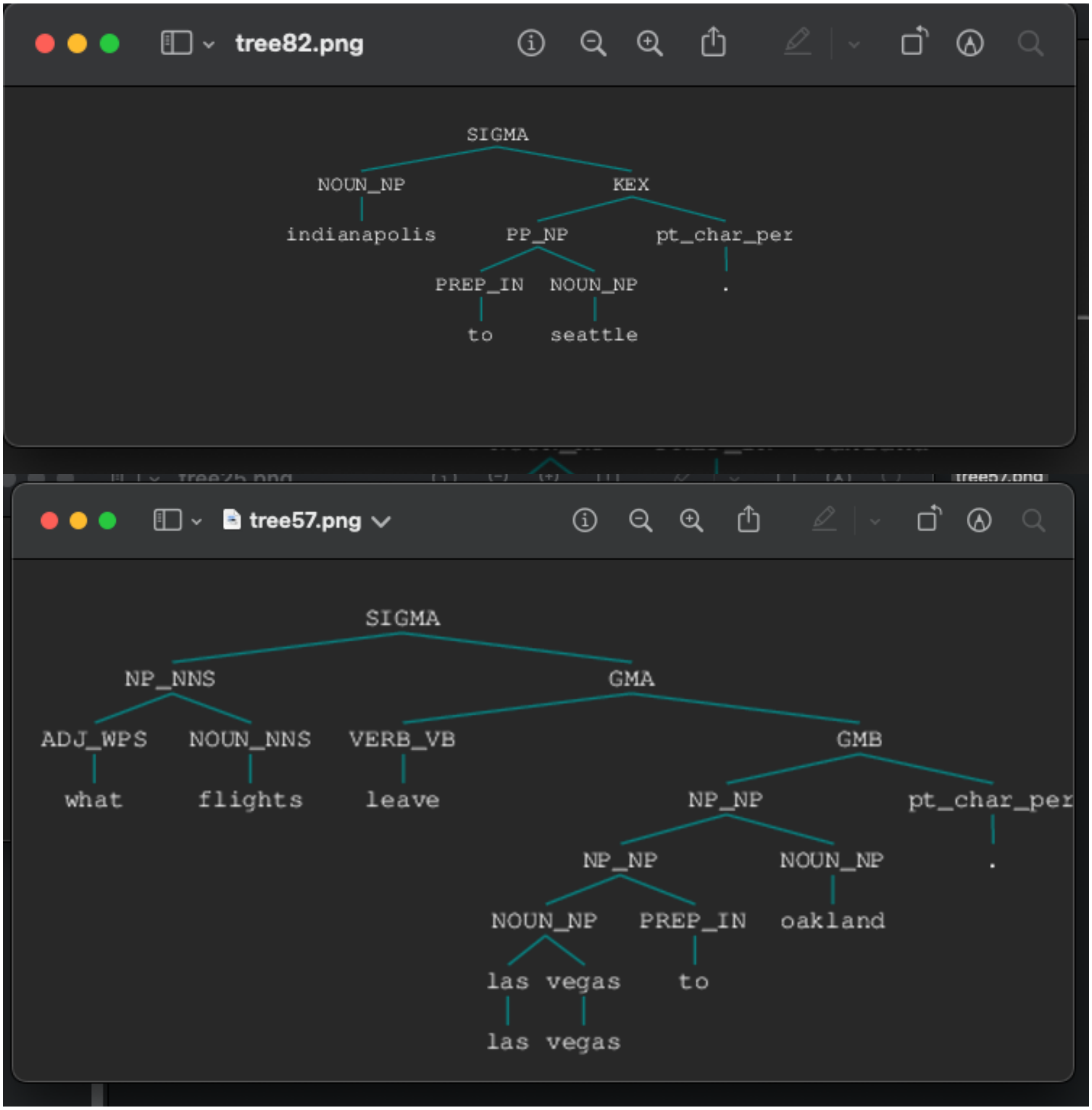


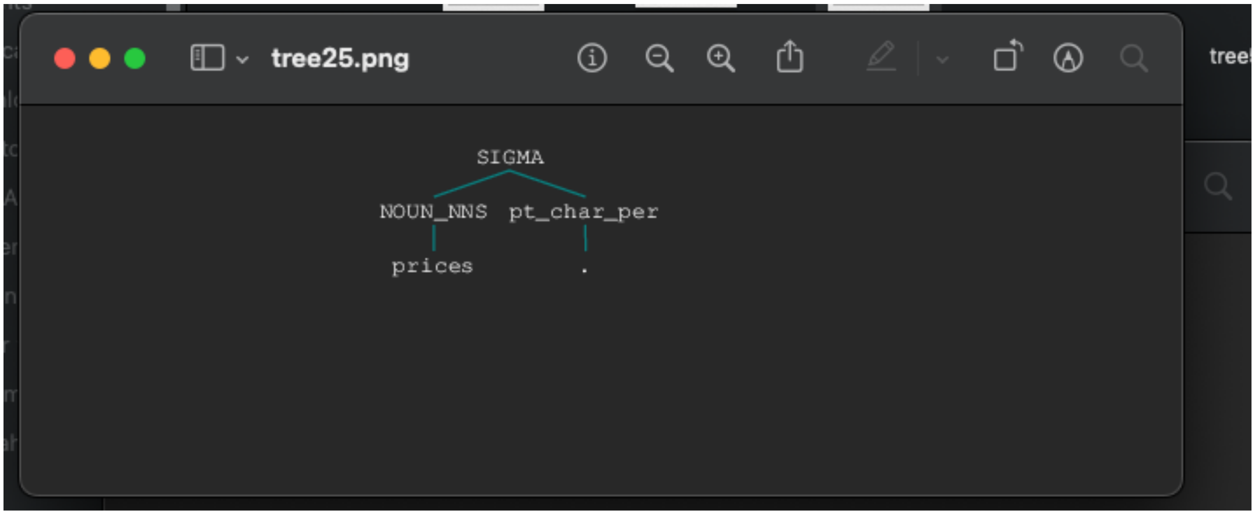
```
                     S
    _________________|____
   |                      PP
   |                  ____|______________
   |                 |                   PP
   |                 |                   |
   |                from                 at
   |                 |                   |
   NP                NP                  NP
   |                 |                   |
Flights             Los                  3
   |        _________|____________       |
  ...   Angeles  to New  York departing  PM


==================================================
```

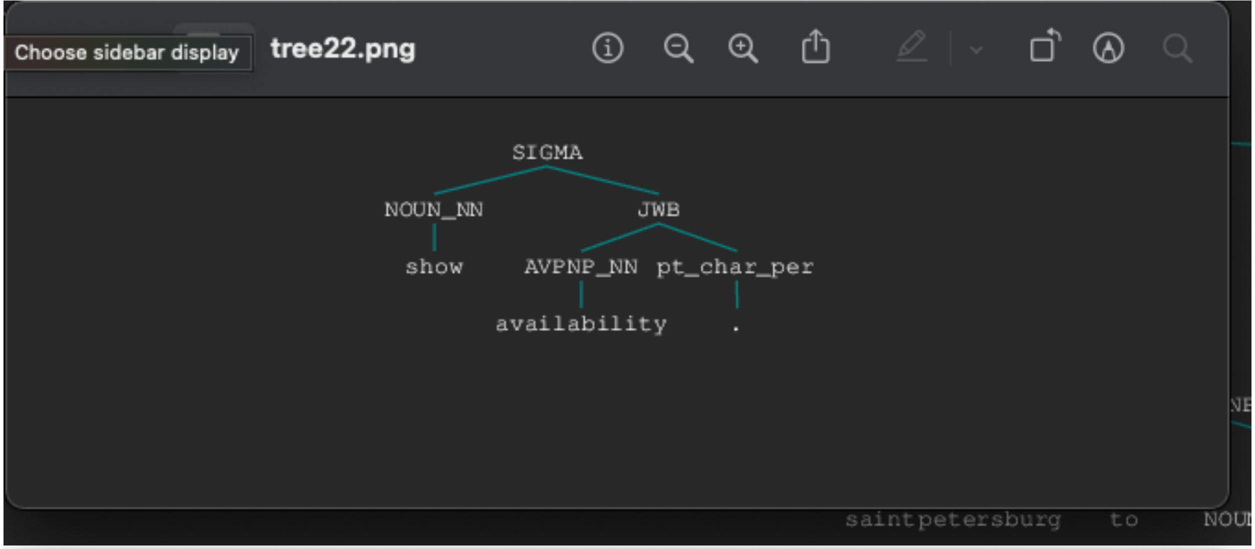
Parse Trees Generated by the Parser:

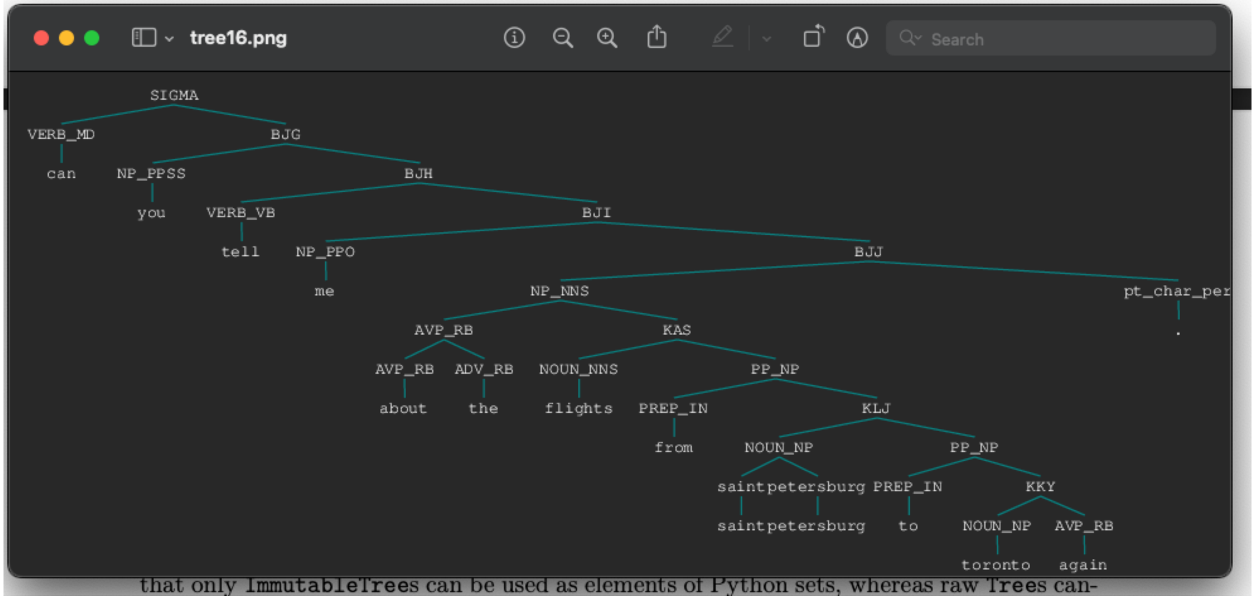





tree82.png
SIGMA
NOUN_NP
KEX
indianapolis
PP_NP
pt_char_per
PREP_IN
NOUN_NP
.
to
seattle
tree57.png
SIGMA
NP_NNS
GMA
ADJ_WPS
NOUN_NNS
VERB_VB
GMB
what
flights
leave
NP_NP
pt_char_per
NP_NP
NOUN_NP
.
NOUN_NP
PREP_IN
oakland
las vegas
to
las
vegas







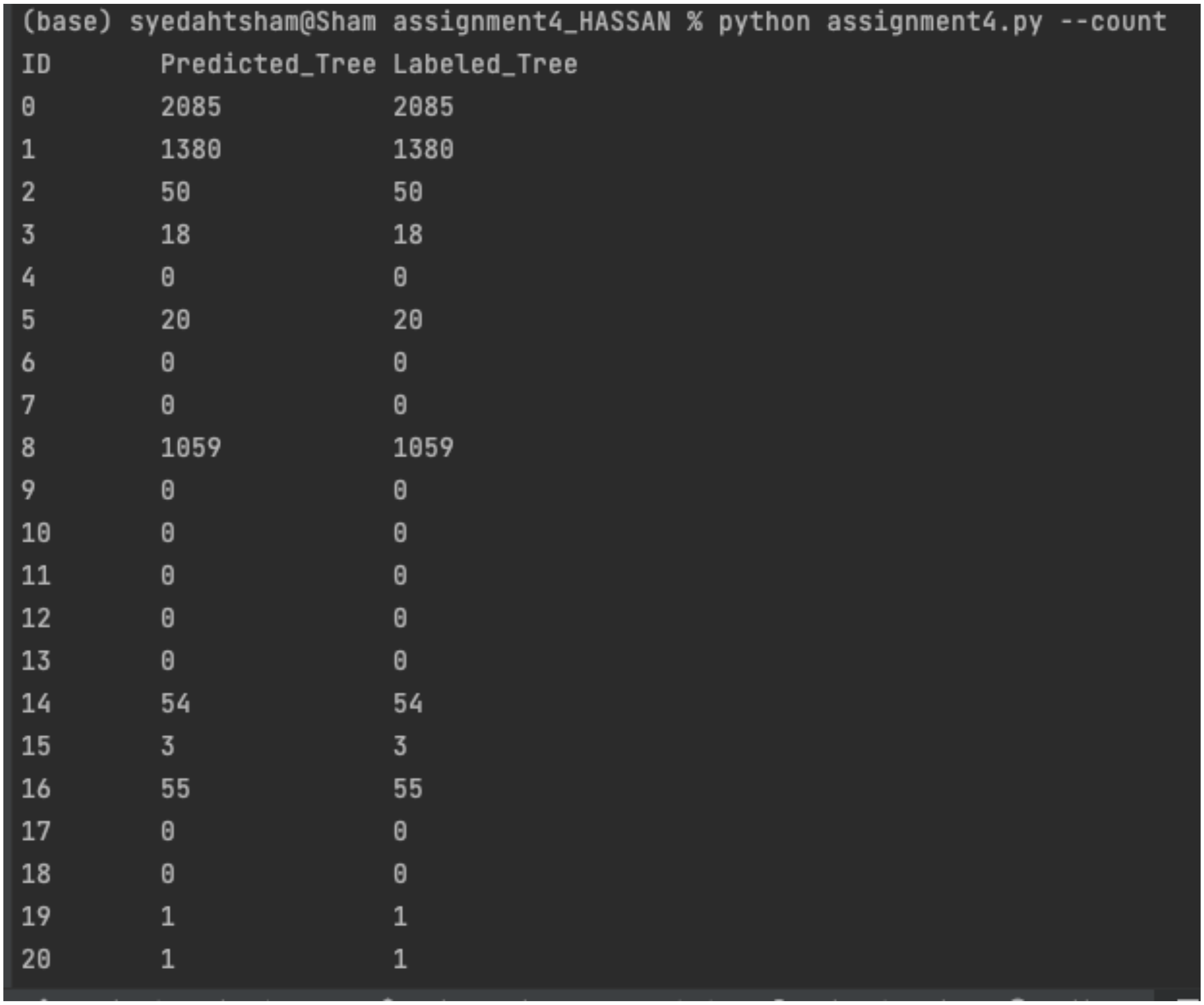
that only ImmutableTrees can be used as elements of Python sets, whereas raw Trees can-

Count Method:

The count method in the CKY parser optimally calculates the number of parse trees for a given syntactic structure in the chart with backpointers, without explicitly generating and storing individual trees. It leverages the information encoded in the backpointers to efficiently compute the total count of valid parse trees, contributing to improved computational efficiency compared to the traditional approach.

Verification of Results:

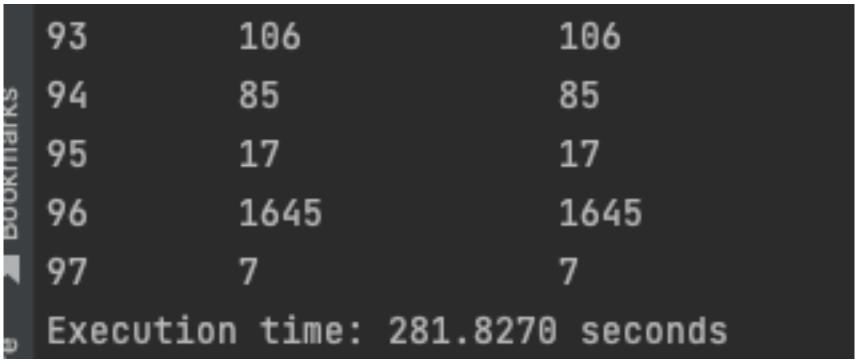
We can see that the predicted number of parse trees and actual number of parse trees are same for each sentence in the test set. Therefore, the count method is computing correctly without even creating parse trees.



```
(base) syedahtsham@Sham assignment4_HASSAN % python assignment4.py --count
ID       Predicted_Tree Labeled_Tree
0        2085           2085
1        1380           1380
2        50             50
3        18             18
4        0              0
5        20             20
6        0              0
7        0              0
8        1059           1059
9        0              0
10       0              0
11       0              0
12       0              0
13       0              0
14       54             54
15       3              3
16       55             55
17       0              0
18       0              0
19       1              1
20       1              1
```

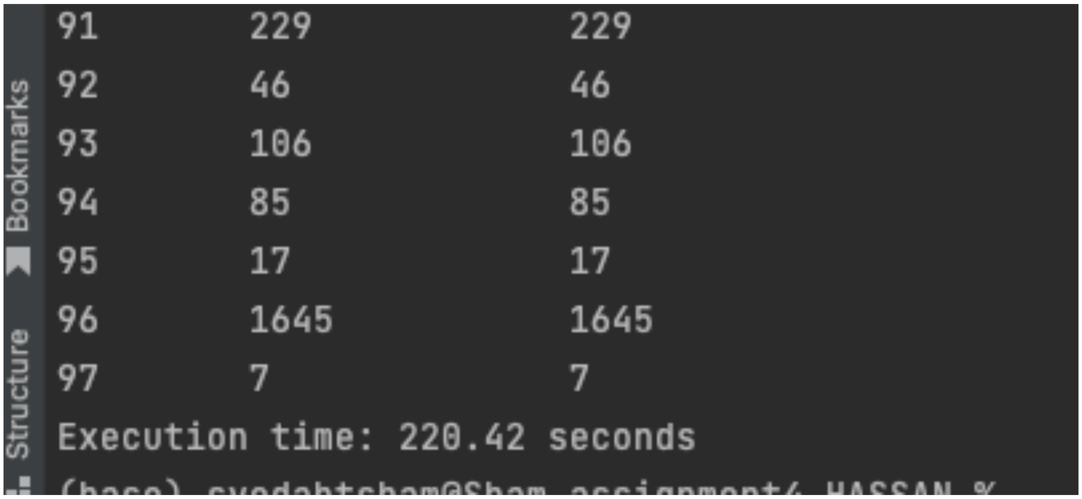
Efficiency of Count vs Parse:

If we use the Parse method, the execution time is around 281 seconds as can be seen below.



```
93       106            106
94       85             85
95       17             17
96       1645           1645
97       7              7
Execution time: 281.8270 seconds
```

And if we use the count method, then execution time is around 220 seconds.



```
91       229            229
92       46             46
93       106            106
94       85             85
95       17             17
96       1645           1645
97       7              7
Execution time: 220.42 seconds
(base) syedahtsham@Sham assignment4_HASSAN %
```

Conclusion:

The observed difference in execution time between the count method (220 seconds) and the Parse method (280 seconds) suggests that the optimized counting procedure is more efficient than the full parse tree generation approach. Here are potential reasons for this:

Efficiency of Counting Method:

The count method computes the number of parse trees without explicitly generating and storing individual trees. It likely involves a more streamlined algorithm that directly calculates the count from the backpointers, leading to improved efficiency.

Reduced Memory Overhead:

Generating and storing parse trees requires substantial memory overhead, especially when dealing with a large number of parses. The count method, by avoiding the construction of full parse trees, reduces memory usage, contributing to faster execution.

Computational Complexity:

Parsing involves constructing complete parse trees, which can be computationally expensive, especially when there are numerous parses. The count method, focusing solely on the count, might involve less complex computations, resulting in quicker execution.

Algorithmic Differences:

The count method may utilize a more optimized algorithm for calculating counts, taking advantage of specific characteristics of the parsing problem. In contrast, the Parse method generates and stores complete parse trees, involving more extensive computations.

Data Characteristics:

The efficiency gains of the counting method might be more pronounced when dealing with certain types of sentences or syntactic structures. The nature of the ATIS dataset and the characteristics of the sentences being parsed can influence the relative performance of the two methods.

In conclusion, the observed difference in execution time suggests that the count method, focusing on counting without full parse tree generation, is a more efficient approach for the given parsing task, demonstrating the benefits of optimizing the algorithm for specific requirements.