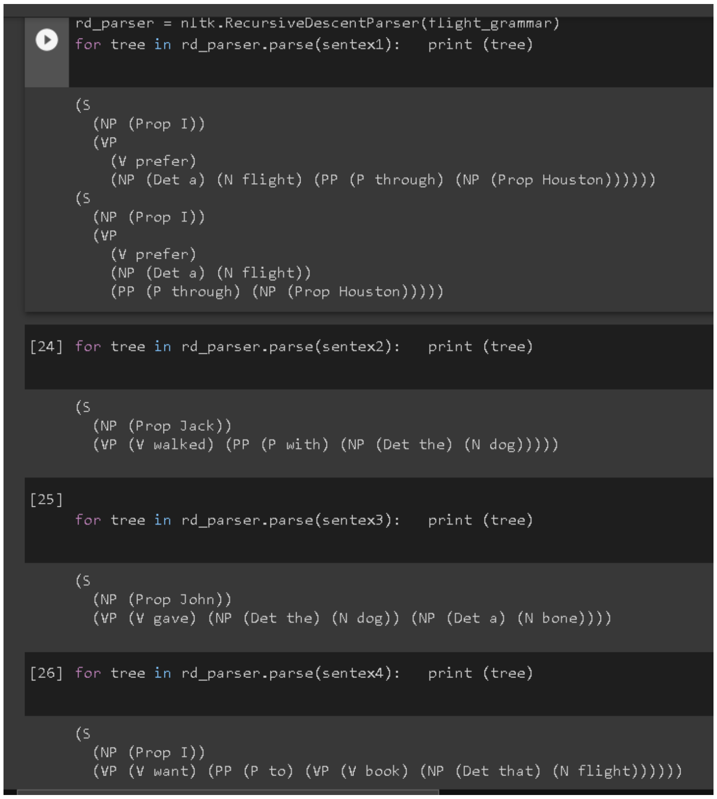
LAB 5
Name:Satwik Hosamani
SUID:378000248

Results:



```
rd_parser = nltk.RecursiveDescentParser(flight_grammar)
for tree in rd_parser.parse(sentex1):   print (tree)

(S
  (NP (Prop I))
  (VP
    (V prefer)
    (NP (Det a) (N flight) (PP (P through) (NP (Prop Houston))))))
(S
  (NP (Prop I))
  (VP
    (V prefer)
    (NP (Det a) (N flight))
    (PP (P through) (NP (Prop Houston)))))

[24] for tree in rd_parser.parse(sentex2):   print (tree)

(S
  (NP (Prop Jack))
  (VP (V walked) (PP (P with) (NP (Det the) (N dog)))))

[25]
     for tree in rd_parser.parse(sentex3):   print (tree)

(S
  (NP (Prop John))
  (VP (V gave) (NP (Det the) (N dog)) (NP (Det a) (N bone))))

[26] for tree in rd_parser.parse(sentex4):   print (tree)

(S
  (NP (Prop I))
  (VP (V want) (PP (P to) (VP (V book) (NP (Det that) (N flight))))))
```

Observations:

1. The rules gave a good understanding of how the traditional parsing algorithm works.
2. All the sentences have to examined and understood before making changes to the grammar
3. The parser predicts the input based on the grammar rules provided by the programmer rather than parsing through the context.
4. The basic rules don't change but the grammar can end up being complex because of the complex sentence structure in english language.

Lessons:

While writing the grammar I identified some of the words in wrong parts of speech which lead to wrong rules.