Part 1

In [85]:
```
from grammar import *
```

In [86]:
```
with open('atis3.pcfg','r') as grammar_file:
    grammar = Pcfg(grammar_file)
```

In [87]:
```
grammar.startsymbol
```

Out[87]:
```
'TOP'
```

In [88]:
```
grammar.lhs_to_rules['PP']
```

Out[88]:
```
[('PP', ('ABOUT', 'NP'), 0.00133511348465),
 ('PP', ('ADVP', 'PPBAR'), 0.00133511348465),
 ('PP', ('AFTER', 'NP'), 0.0253671562083),
 ('PP', ('AROUND', 'NP'), 0.00667556742323),
 ('PP', ('AS', 'ADJP'), 0.00133511348465),
 ('PP', ('AT', 'NP'), 0.0106809078772),
 ('PP', ('AT', 'PPBAR'), 0.00133511348465),
 ('PP', ('BEFORE', 'NP'), 0.0146862483311),
 ('PP', ('BETWEEN', 'NP'), 0.0160213618158),
 ('PP', ('BY', 'NP'), 0.00267022696929),
 ('PP', ('DURING', 'NP'), 0.00133511348465),
 ('PP', ('FOR', 'NP'), 0.00934579439252),
 ('PP', ('FROM', 'NP'), 0.332443257677),
 ('PP', ('IN', 'NP'), 0.0587449933244),
 ('PP', ('INTO', 'NP'), 0.00133511348465),
 ('PP', ('NP', 'PPBAR'), 0.00133511348465),
 ('PP', ('OF', 'NP'), 0.0520694259012),
 ('PP', ('ON', 'NP'), 0.110814419226),
 ('PP', ('PP', 'PP'), 0.00133511348465),
 ('PP', ('PP', 'PPBAR'), 0.00133511348465),
 ('PP', ('THAN', 'NP'), 0.00267022696929),
 ('PP', ('THROUGH', 'NP'), 0.00133511348465),
 ('PP', ('TO', 'ADVP'), 0.00133511348465),
 ('PP', ('TO', 'INTJ'), 0.00133511348465),
 ('PP', ('TO', 'NP'), 0.324432576769),
 ('PP', ('VIA', 'NP'), 0.00267022696929),
 ('PP', ('WITH', 'NP'), 0.00934579439252),
 ('PP', ('WITHIN', 'NP'), 0.00267022696929),
 ('PP', ('WITHOUT', 'NP'), 0.00133511348465),
 ('PP', ('from',), 0.00133511348465)]
```

In [89]:
```
grammar.rhs_to_rules[('ABOUT','NP')]
```

Out[89]:
```
[('PP', ('ABOUT', 'NP'), 0.00133511348465)]
```

In [90]:
```
grammar.rhs_to_rules[('NP','VP')]
```

Out[90]:
```
[('NP', ('NP', 'VP'), 0.00602409638554),
 ('S', ('NP', 'VP'), 0.694915254237),
 ('SBAR', ('NP', 'VP'), 0.166666666667),
 ('SQBAR', ('NP', 'VP'), 0.289156626506)]
```

In [91]:
```
!python grammar.py atis3.pcfg
```

```
The grammar is valid PCFG in CNF!
```

Part-2

In [92]:
```
from cky import *
```

In [93]:
```
 parser = CkyParser(grammar)
```

In [94]:
```
toks =['flights', 'from', 'miami', 'to', 'cleveland', '.']
```

In [95]:
```
parser.is_in_language(toks)
```

Out[95]:
```
True
```

In [96]:
```
 toks =['miami', 'flights','cleveland', 'from', 'to','.']
```

In [97]:
```
parser.is_in_language(toks)
```

Out[97]:
```
False
```

Part 3

In [98]:
```
parser = CkyParser(grammar)
```

In [99]:
```
toks =['flights', 'from', 'miami', 'to', 'cleveland', '.']
```

In [100]:
```
table, probs = parser.parse_with_backpointers(toks)
```

In [101]:
```
check_table_format(table)
```

Out[101]:
```
True
```

In [102]:
```
check_probs_format(probs)
```

Out[102]:
```
True
```

Part 4

In [103]:
```
 parser = CkyParser(grammar)
```

In [104]:
```
toks =['flights', 'from', 'miami', 'to', 'cleveland', '.']
```

In [105]:
```
table, probs = parser.parse_with_backpointers(toks)
```

In [106]:
```
get_tree(table, 0, len(toks), grammar.startsymbol)
```

Out[106]:
```
('TOP',
 ('NP',
  ('NP', 'flights'),
  ('NPBAR',
   ('PP', ('FROM', 'from'), ('NP', 'miami')),
   ('PP', ('TO', 'to'), ('NP', 'cleveland')))),
 ('PUN', '.'))
```

In [107]:
```
!python cky.py
```

```
True
```

Part 5

In [108]:
```
!python evaluate_parser.py atis3.pcfg atis3_test.ptb
```

```
target:     ('TOP', ('WHNP', ('WHNP', ('WHAT', 'what'), ('FLIGHTS', 'flights')), ('WHNPBAR', ('PP', ('FROM', 'from'), ('NP', ('KANSAS', 'kansas'), ('CITY', 'city'))), ('PP', ('T
O', 'to'), ('NP', 'denver')))), ('PUN', '.'))
predicted:  ('TOP', ('WHNP', ('WHNP', ('WHAT', 'what'), ('FLIGHTS', 'flights')), ('WHNPBAR', ('PP', ('FROM', 'from'), ('NP', ('KANSAS', 'kansas'), ('CITY', 'city'))), ('PP', ('T
O', 'to'), ('NP', 'denver')))), ('PUN', '.'))
P:1.0 R:1.0 F:1.0

input:  ['what', 'flights', 'from', 'minneapolis', 'to', 'pittsburgh', '.']
target:     ('TOP', ('WHNP', ('WHNP', ('WHAT', 'what'), ('FLIGHTS', 'flights')), ('WHNPBAR', ('PP', ('FROM', 'from'), ('NP', 'minneapolis')), ('PP', ('TO', 'to'), ('NP', 'pittsbur
gh')))), ('PUN', '.'))
predicted:  ('TOP', ('WHNP', ('WHNP', ('WHAT', 'what'), ('FLIGHTS', 'flights')), ('WHNPBAR', ('PP', ('FROM', 'from'), ('NP', 'minneapolis')), ('PP', ('TO', 'to'), ('NP', 'pittsbur
gh')))), ('PUN', '.'))
P:1.0 R:1.0 F:1.0

input:  ['what', 'flights', 'from', 'tampa', 'to', 'cincinnati', '.']
target:     ('TOP', ('WHNP', ('WHNP', ('WHAT', 'what'), ('FLIGHTS', 'flights')), ('WHNPBAR', ('PP', ('FROM', 'from'), ('NP', 'tampa')), ('PP', ('TO', 'to'), ('NP', 'cincinnat
i')))), ('PUN', '.'))
predicted:  ('TOP', ('WHNP', ('WHNP', ('WHAT', 'what'), ('FLIGHTS', 'flights')), ('WHNPBAR', ('PP', ('FROM', 'from'), ('NP', 'tampa')), ('PP', ('TO', 'to'), ('NP', 'cincinnat
i')))), ('PUN', '.'))
P:1.0 R:1.0 F:1.0
```

In [ ]: