

```
In [1]: from grammar import *
        from parser import *
        from util2 import *
```

The given PCFG:

```
In [2]: print str(timeFliesPCFG)

Noun => flies      | 0.4
Noun => arrow      | 0.4
Noun => time       | 0.2
TOP  => S          | 1.0
Det  => an         | 1.0
VP   => Verb NP    | 0.6
VP   => Verb PP    | 0.2
VP   => Verb       | 0.1
VP   => Verb NP_PP | 0.1
S    => VP | 0.2
S    => NP VP      | 0.5
S    => VP PP      | 0.1
S    => NP VP_PP   | 0.2
VP_PP => VP PP    | 1.0
NP_PP => NP PP    | 1.0
Prep => like      | 1.0
PP   => Prep NP   | 1.0
Verb => flies     | 0.5
Verb => like      | 0.2
Verb => time      | 0.3
NP   => Det Noun  | 0.7
NP   => Noun      | 0.3
```

The sentence we wish to parse:

```
In [3]: print timeFliesSent

['time', 'flies', 'like', 'an', 'arrow']
```

Expected output:

```
(TOP:
  (S:
    (NP: (Noun: 'time'))
    (VP:
      (Verb: 'flies')
      (PP: (Prep: 'like') (NP: (Det: 'an') (Noun: 'arrow'))))))
```

Actual output:

```
In [4]: print parse(timeFliesPCFG, timeFliesSent)

(TOP:
  (S:
    (NP: (Noun: 'time'))
    (VP:
      (Verb: 'flies')
      (PP: (Prep: 'like') (NP: (Det: 'an') (Noun: 'arrow'))))))
```

The chart produced:

```
In [5]: chart = cky(timeFliesPCFG, timeFliesSent, pruningPercent=None)
        printChart(chart,timeFliesSent, widths=(5,11,7,3,2), printBackPointers=False, printProbs=False)

row0 _____
```

```

1
Noun
TOP
VP
S
Verb
time
NP
2
VP
S
TOP
3
4
5
VP
S
VP_PP
TOP

row1_____
1
2
flies
Noun
TOP
VP
S
Verb
NP
3
S
TOP
4
5
VP
NP_PP
S
VP_PP
TOP

row2_____
1
2
3
like
TOP
VP
S
Verb
Prep
4
5
VP
PP
S

row3_____
1
2
3
4
Det
an
5
NP

row4_____
1
2
3

```

4
5
NP
Noun
arrow

```
In [6]: printChart(chart,timeFliesSent, widths=(5,11,7,3,2), printBackPointers=True)
```



