In [2]:

```
import nltk
from nltk.corpus import state_union
from pprint import pprint
```

In [3]:

```
from nltk.corpus import conll2000
train_sents = conll2000.chunked_sents('train.txt')
test_sents = conll2000.chunked_sents('test.txt')
```

In [4]:

```
pprint(train_sents)
```

```
[Tree('S', [Tree('NP', [('Confidence', 'NN')]), Tree('PP', [('in', 'IN')]), Tree('N
P', [('the', 'DT'), ('pound', 'NN')]), Tree('VP', [('is', 'VBZ'), ('widely', 'RB'),
('expected', 'VBN'), ('to', 'TO'), ('take', 'VB')]), Tree('NP', [('another', 'DT'),
('sharp', 'JJ'), ('dive', 'NN')]), ('if', 'IN'), Tree('NP', [('trade', 'NN'), ('figu
res', 'NNS')]), Tree('PP', [('for', 'IN')]), Tree('NP', [('September', 'NNP')]),
(',', ','), ('due', 'JJ'), Tree('PP', [('for', 'IN')]), Tree('NP', [('release', 'N
N')]), Tree('NP', [('tomorrow', 'NN')]), (',', ','), Tree('VP', [('fail', 'VB'), ('t
o', 'TO'), ('show', 'VB')]), Tree('NP', [('a', 'DT'), ('substantial', 'JJ'), ('impro
vement', 'NN')]), Tree('PP', [('from', 'IN')]), Tree('NP', [('July', 'NNP'), ('and',
'CC'), ('August', 'NNP')]), Tree('NP', [("'s", 'POS'), ('near-record', 'JJ'), ('defi
cits', 'NNS')]), ('.', '.')]), Tree('S', [('Chancellor', 'NNP'), Tree('PP', [('of',
'IN')]), Tree('NP', [('the', 'DT'), ('Exchequer', 'NNP')]), Tree('NP', [('Nigel', 'N
NP'), ('Lawson', 'NNP')]), Tree('NP', [("'s", 'POS'), ('restated', 'VBN'), ('commitm
ent', 'NN')]), Tree('PP', [('to', 'TO')]), Tree('NP', [('a', 'DT'), ('firm', 'NN'),
('monetary', 'JJ'), ('policy', 'NN')]), Tree('VP', [('has', 'VBZ'), ('helped', 'VB
N'), ('to', 'TO'), ('prevent', 'VB')]), Tree('NP', [('a', 'DT'), ('freefall', 'N
N')]), Tree('PP', [('in', 'IN')]), Tree('NP', [('sterling', 'NN')]), Tree('PP', [('o
ver', 'IN')]), Tree('NP', [('the', 'DT'), ('past', 'JJ'), ('week', 'NN')]), ('.',
'.')]), ...]
```

In [5]:

```
pprint(test_sents)
```

```
[Tree('S', [Tree('NP', [('Rockwell', 'NNP'), ('International', 'NNP'), ('Corp.', 'NN
P')]), Tree('NP', [("'s", 'POS'), ('Tulsa', 'NNP'), ('unit', 'NN')]), Tree('VP',
[('said', 'VBD')]), Tree('NP', [('it', 'PRP')]), Tree('VP', [('signed', 'VBD')]), Tr
ee('NP', [('a', 'DT'), ('tentative', 'JJ'), ('agreement', 'NN')]), Tree('VP', [('ext
ending', 'VBG')]), Tree('NP', [('its', 'PRP$'), ('contract', 'NN')]), Tree('PP',
[('with', 'IN')]), Tree('NP', [('Boeing', 'NNP'), ('Co.', 'NNP')]), Tree('VP', [('t
o', 'TO'), ('provide', 'VB')]), Tree('NP', [('structural', 'JJ'), ('parts', 'NN
S')]), Tree('PP', [('for', 'IN')]), Tree('NP', [('Boeing', 'NNP')]), Tree('NP',
[("'s", 'POS'), ('747', 'CD'), ('jetliners', 'NNS')]), ('.', '.')]), Tree('S', [Tree
('NP', [('Rockwell', 'NNP')]), Tree('VP', [('said', 'VBD')]), Tree('NP', [('the', 'D
T'), ('agreement', 'NN')]), Tree('VP', [('calls', 'VBZ')]), ('for', 'IN'), Tree('N
P', [('it', 'PRP')]), Tree('VP', [('to', 'TO'), ('supply', 'VB')]), Tree('NP', [('20
0', 'CD'), ('additional', 'JJ'), ('so-called', 'JJ'), ('shipsets', 'NNS')]), Tree('P
P', [('for', 'IN')]), Tree('NP', [('the', 'DT'), ('planes', 'NNS')]), ('.', '.')]),
...]
```

In [6]:

```
file1 = train_sents[0]
```

In [8]:

```
file2 = test_sents[0]
```

In [9]:

```
pattern = """NP: {<DT>?<JJ>*<NN>}"""
chunk_parser = nltk.RegexpParser(pattern)
print(chunk_parser.parse(file1))
```

```
(S
  (NP Confidence/NN)
  (PP in/IN)
  (NP the/DT pound/NN)
  (VP is/VBZ widely/RB expected/VBN to/TO take/VB)
  (NP another/DT sharp/JJ dive/NN)
  if/IN
  (NP trade/NN figures/NNS)
  (PP for/IN)
  (NP September/NNP)
  ,/,
  due/JJ
  (PP for/IN)
  (NP release/NN)
  (NP tomorrow/NN)
  ,/,
  (VP fail/VB to/TO show/VB)
  (NP a/DT substantial/JJ improvement/NN)
  (PP from/IN)
  (NP July/NNP and/CC August/NNP)
  (NP 's/POS near-record/JJ deficits/NNS)
  ./.)
```

In [10]:
```python
results = chunk_parser.parse(file1)
print(results)
```

```
(S
  (NP Confidence/NN)
  (PP in/IN)
  (NP the/DT pound/NN)
  (VP is/VBZ widely/RB expected/VBN to/TO take/VB)
  (NP another/DT sharp/JJ dive/NN)
  if/IN
  (NP trade/NN figures/NNS)
  (PP for/IN)
  (NP September/NNP)
  ,/,
  due/JJ
  (PP for/IN)
  (NP release/NN)
  (NP tomorrow/NN)
  ,/,
  (VP fail/VB to/TO show/VB)
  (NP a/DT substantial/JJ improvement/NN)
  (PP from/IN)
  (NP July/NNP and/CC August/NNP)
  (NP 's/POS near-record/JJ deficits/NNS)
  ./.)
```

In [13]:
```python
metrics = chunk_parser.evaluate(test_sents)
```

In [14]:
```python
print("Chunking accuracy:")
print("Precision:", metrics.precision())
print("Recall:", metrics.recall())
print("F1-score:", metrics.f_measure())
```

```
Chunking accuracy:
Precision: 0.4531767539897621
Recall: 0.13749942898908227
F1-score: 0.21098377317492026
```

In [ ]:

```

```