```python
import nltk
from nltk import CFG

from nltk.parse import RecursiveDescentParser
from nltk.parse import ShiftReduceParser

myg=CFG.fromstring("""
S -> NP VP
PP -> P NP
NP -> "the" N | N PP | "the" N PP
VP -> V NP | V PP | V NP PP
N -> "cat" | "dog" | "rug"
V -> "chased" | "sat"
P -> "in" | "on"
""")

rd=RecursiveDescentParser(myg)

sent1="the cat chased the dog".split()
sent2="the cat chased the dog on the rug".split()

print('\nRecursive Descent Parser')
print('\nthe cat chased the dog\n')
for t in rd.parse(sent1):
    print(t)
print('\nthe cat chased the dog on the rug\n')
for t in rd.parse(sent2):
    print(t)

srp=ShiftReduceParser(myg)

print('\nShift Reduce Parser')
print('\nthe cat chased the dog\n')
for t in srp.parse(sent1):
    print(t)
print('\nthe cat chased the dog on the rug\n')
for t in rd.parse(sent2):
    print(t)
#**************************************************************
from nltk import CFG
import nltk
from nltk.parse import ChartParser

myg=CFG.fromstring("""
S -> NP VP
PP -> P NP
NP -> Det N | Det N PP | 'I'
VP -> V NP | VP PP
Det -> 'an' | 'my'
N -> 'elephant' | 'pajamas'
V -> 'shot'
P -> 'in'
""")

sent="I shot an elephant in my pajamas".split()
cp=ChartParser(myg)
print('\nChart Parser')
for t in cp.parse(sent):
    print(t)
```