

## بخش ۲

در این بخش تابعی نوشته شده است که در آن گرامر به همراه جمله و نوع پارسر داده می‌شود و در آن درخت‌ها نمایش داده می‌شود. (هر درخت در یک پنجره‌ی جدید) حالت پیش فرض نوع پارسر ChatParser است.

## part 2 : draw parse trees

```
In [2]: 1 def show_trees(grammar,sentence , parser_type=nltk.ChartParser):
2         tokenizer = nltk.tokenize.RegexpTokenizer('\w+')
3         tokens = tokenizer.tokenize(sentence)
4         parser = parser_type(grammar)
5         for tree in parser.parse(tokens):
6             tree.draw() # show in a new window
7             print(tree)
8
9
```

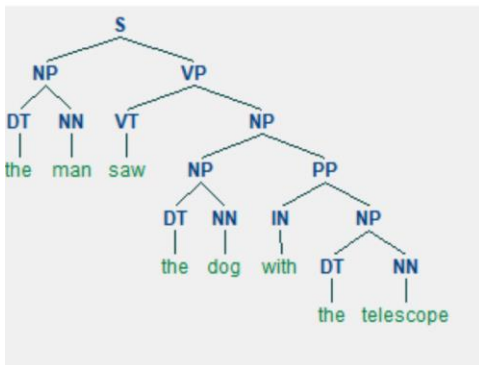
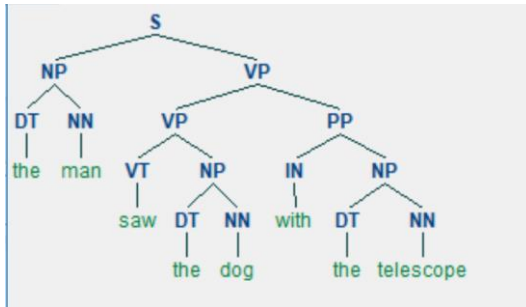
## بخش ۳

در این بخش فایل Grammar1.cfg خوانده شده و تابع show\_trees برای آن و جمله‌ی داده شده صدا زده شده است.

## part 3 : grammar 1

```
In [3]: 1 grammar = nltk.data.load("Grammar1.cfg")
2         show_trees(grammar,"the man saw the dog with the telescope")
```

درخت‌های ساخته شده به صورت زیر است.



همانطور که نمایش داده شده است، این جمله دو درخت تجزیه دارد. دلیل ایجاد دو درخت، ابهام این جمله است و این که می‌تواند به دو صورت برای آن با گرامر داده شده تجزیه کرد.

در مفهوم هم می‌توان دو صورت خواند:

- مردی که سگ را دید تلسکوپ دارد
- مردی سگی را دید که تلسکوپ دارد.

## بخش ۴

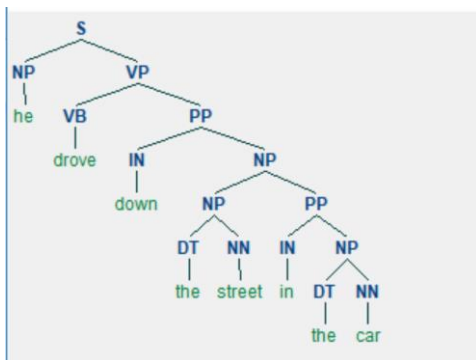
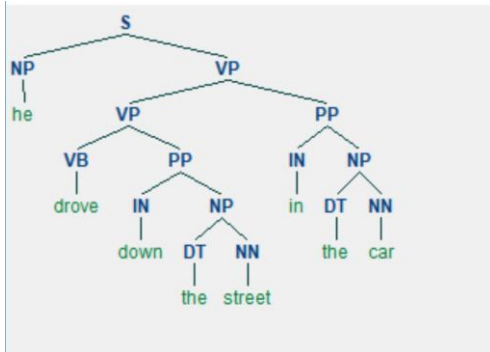
در این بخش گرامرهای گفته شده در فایل متناظر نوشته شده است و هر کدام جداگانه خوانده شده و درخت‌ها نمایش داده می‌شود.

## گرامر ۲

### for grammar 2

```
In [6]: 1 grammar = nltk.data.load("Grammar2.cfg")
        2 show_trees(grammar,"he drove down the street in the car")
```

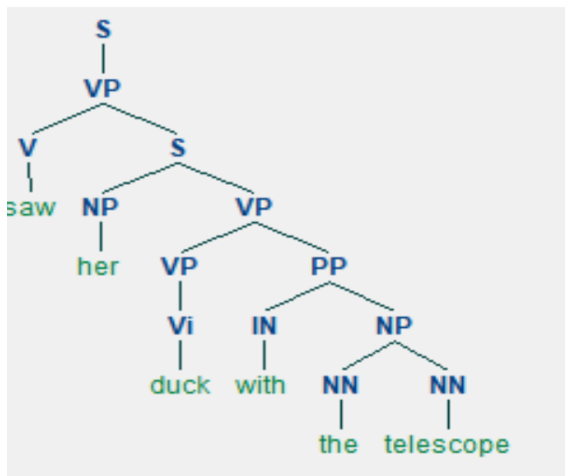
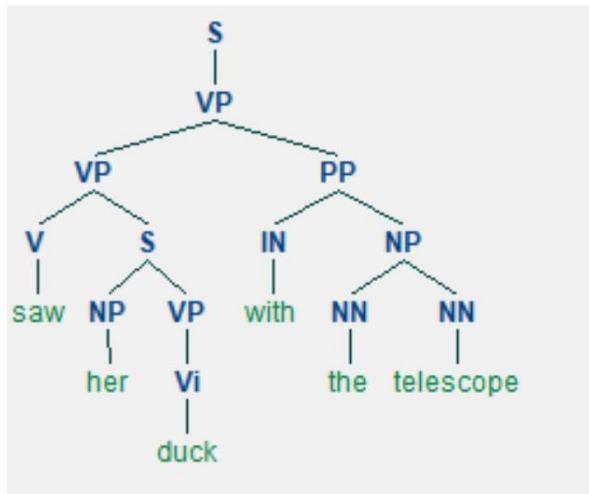
درخت‌های آن به صورت زیر است:



گرامر ۳

for grammar 3

```
In [7]: 1 grammar = nltk.data.load("Grammar3.cfg")
        2 show_trees(grammar,"saw her duck with the telescope")
```

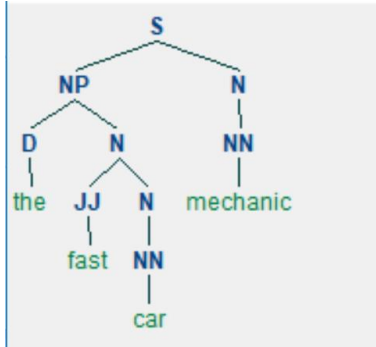


گرامر ۴

## for grammar 4

```

In [8]: 1 grammar = nltk.data.load("Grammar4.cfg")
        2 show_trees(grammar,"the fast car mechanic")
  
```

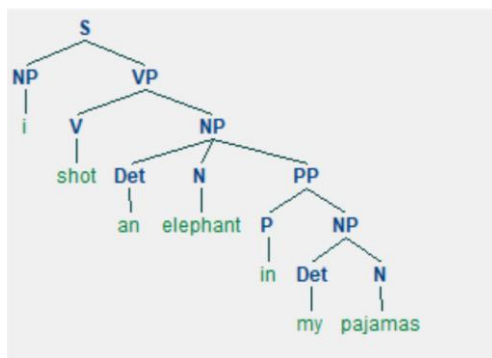
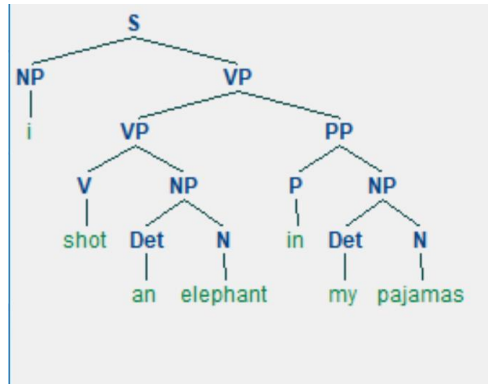


## بخش ۵

کد مربوط به این بخش به صورت زیر است. (گرامر به صورت استرینگ خوانده می‌شود و تابع نوشته شده صدا زده می‌شود).

## part 5

```
In [9]: 1 grammar = nltk.CFG.fromstring("""
2 S -> NP VP
3 PP -> P NP
4 NP -> Det N | Det N PP | 'i'
5 VP -> V NP | VP PP
6 Det -> 'an' | 'my'
7 N -> 'elephant' | 'pajamas'
8 V -> 'shot'
9 P -> 'in'
10 """)
11 show_trees(grammar,"i shot an elephant in my pajamas")
```



## بخش ۶

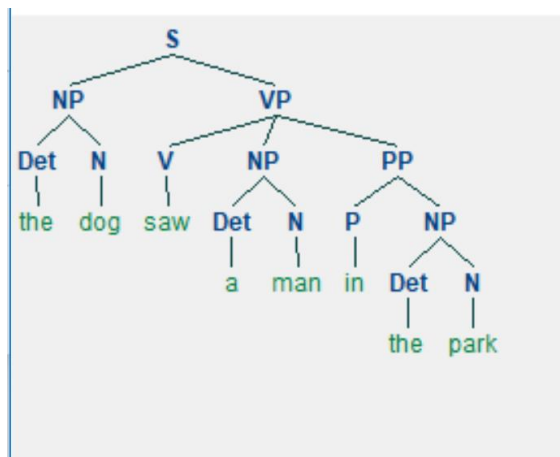
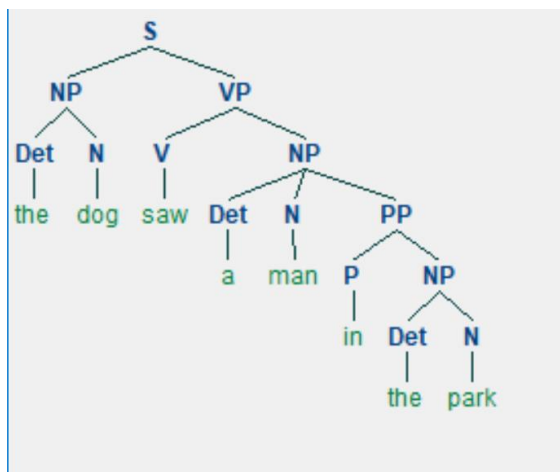
به مشابه بخش قبل کد نوشته شده است با این تفاوت که نوع پارسر RecursiveDescentParser قرار داده شده است.

### part 6

```

In [8]: 1 grammar = nltk.CFG.fromstring("""
2 S -> NP VP
3 VP -> V NP | V NP PP
4 PP -> P NP
5 NP -> Det N | Det N PP
6 Det -> "a" | "the"
7 N -> "man" | "dog" | "park"
8 P -> "in"
9 V -> "saw"
10 """)
11 show_trees(grammar, "the dog saw a man in the park" , parser_type=nltk.parse.RecursiveDescentParser)

```



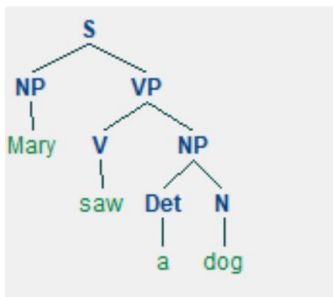
## بخش ۷

به مشابه بخش قبل کد نوشته شده است با این تفاوت که نوع پارسر ShiftReduceParser قرار داده شده است.

## part 7

```

In [11]: 1 grammar = nltk.CFG.fromstring("""
          2 S -> NP VP
          3 VP -> V NP
          4 V -> "saw"
          5 NP -> "Mary" | Det N
          6 Det -> "a"
          7 N -> "dog"
          8 """)
          9 show_trees(grammar,"Mary saw a dog" , parser_type=nltk.parse.ShiftReduceParser)
  
```



## بخش ۸

در روش `InsideChartParser` برای انتخاب گرامر به احتمالات توجه می‌کند ولی در روش `UnsortedChartParser` ، انتخاب گرامرها به صورت تصادفی است. در روش `RandomChartParser` نیز به ترتیب نوشته شده است.

## بخش ۹

### با `InsideChartParser`

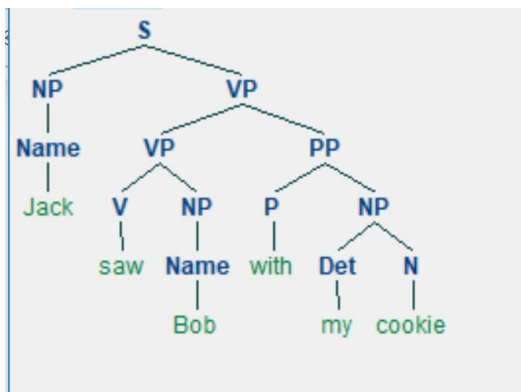
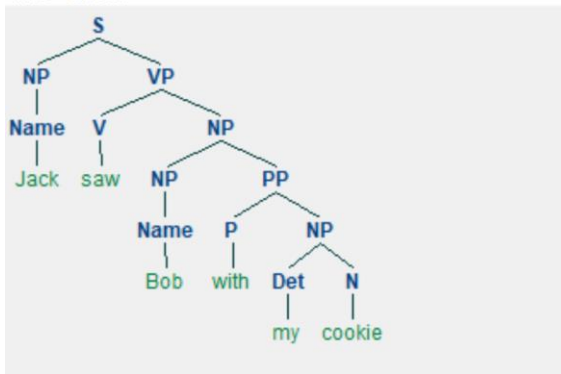
کد به همراه احتمال به صورت زیر است.

#### with `InsideChartParser`

```
In [12]: 1 show_trees(nltk.grammar.toy_pcfg2,"Jack saw|Bob with my cookie", parser_type=nltk.parse.pchart.InsideChartParser)
```

```
(S
  (NP (Name Jack))
  (VP
    (V saw)
    (NP
      (NP (Name Bob))
      (PP (P with) (NP (Det my) (N cookie)))))) (p=6.31607e-06)
(S
  (NP (Name Jack))
  (VP
    (VP (V saw) (NP (Name Bob)))
    (PP (P with) (NP (Det my) (N cookie)))))) (p=2.03744e-07)
```





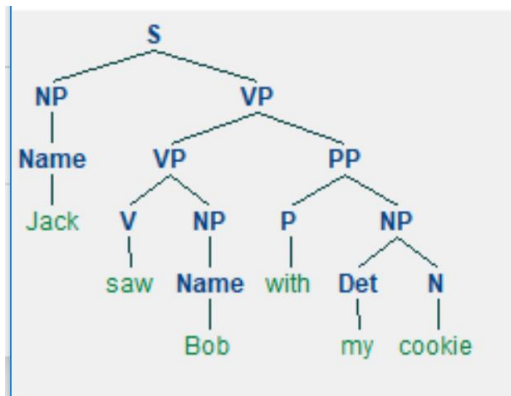
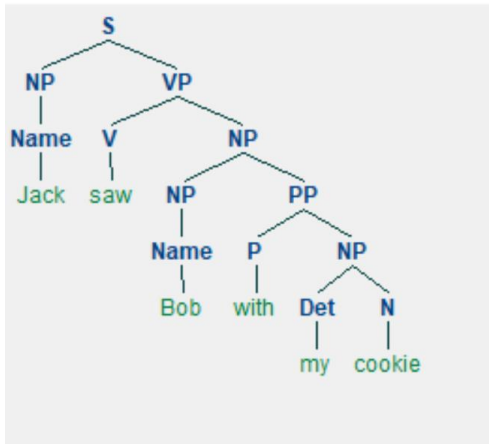
## با RandomChartParser

کد به همراه احتمال به صورت زیر است.

### with RandomChartParser

```
In [13]: 1 show_trees(nltk.grammar.toy_pcfg2, "Jack saw Bob with my cookie", parser_type=nltk.parse.pchart.RandomChartParser)
```

```
(S
  (NP (Name Jack))
  (VP
    (V saw)
    (NP
      (NP (Name Bob))
      (PP (P with) (NP (Det my) (N cookie)))))) (p=6.31607e-06)
(S
  (NP (Name Jack))
  (VP
    (VP (V saw) (NP (Name Bob)))
    (PP (P with) (NP (Det my) (N cookie)))))) (p=2.03744e-07)
```



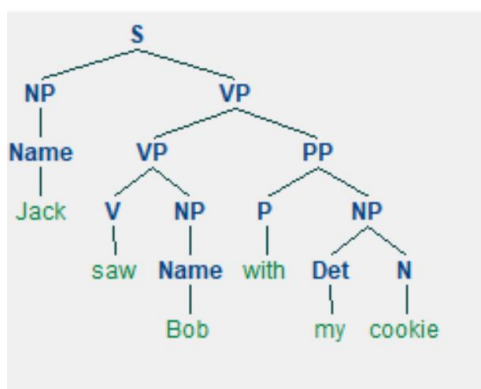
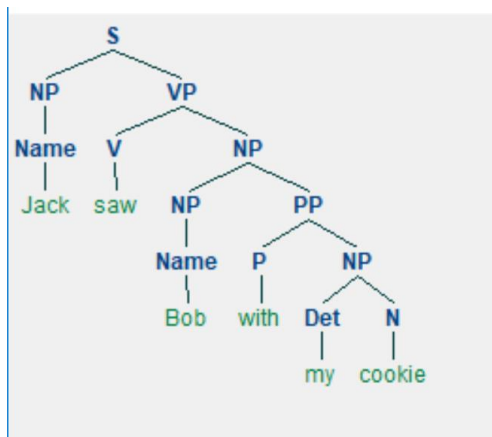
با UnsortedChartParser

کد به همراه احتمال به صورت زیر است.

with UnsortedChartParser

```
In [14]: 1 show_trees(nltk.grammar.toy_pcfg2, "Jack saw Bob with my cookie", parser_type=nltk.parse.pchart.UnsortedChartParser)
```

```
(S
  (NP (Name Jack))
  (VP
    (V saw)
    (NP
      (NP (Name Bob))
      (PP (P with) (NP (Det my) (N cookie)))))) (p=6.31607e-06)
(S
  (NP (Name Jack))
  (VP
    (VP (V saw) (NP (Name Bob)))
    (PP (P with) (NP (Det my) (N cookie)))))) (p=2.03744e-07)
```



بخش ۱۰

قطعه کد و گرامر و احتمال آن به صورت زیر است: (احتمال = ۰.۱۲)

## part 10

```
In [15]: 1 grammar = nltk.PCFG.fromstring("""
2 S -> NP VP [1.0]
3 VP -> Vi [0.3] | Vt NP [0.5] | VP PP [0.2]
4 NP -> DT NN [0.8] | NP PP [0.2]
5 PP -> IN NP [1.0]
6 Vi -> "sleeps" [1.0]
7 Vt -> "saw" [1.0]
8 NN -> "man" [0.1] | "woman" [0.1] | "telescope" [0.3] | "dog" [0.5]
9 DT -> "the" [1.0]
10 IN -> "with" [0.6] | "in" [0.4]
11 """)
12
13 |
14
15 sentence = 'the dog sleeps'
16
17
18 show_trees(grammar, "the dog sleeps", parser_type=nltk.parse.ViterbiParser)
19
```

(S (NP (DT the) (NN dog)) (VP (Vi sleeps))) (p=0.12)

