# Name :Viviyan Richards W

Roll no:205229133

In [1]:
```
import nltk,re,pprint
from nltk.tree import Tree
from nltk.tokenize import word_tokenize
from nltk.tag import pos_tag
from nltk.chunk import ne_chunk
import numpy as npt
```

**Exercise 1**

In [2]:
```
np =nltk.Tree.fromstring('(NP (N Marge))')
np.pretty_print()
```

```
  NP
  |
  N
  |
Marge
```

In [3]:
```
vp = nltk.Tree.fromstring('(VP (V make) (NP (DET a) (N ham) (N sandwich)))')
vp.pretty_print()
```

```
          VP
  ________|___
 |            NP
 |     _______|_____
 V   DET      N     N
 |    |       |     |
make  a      ham sandwich
```

In [4]:
```
aux= nltk.Tree.fromstring('(AUX will)')
aux.pretty_print()
```

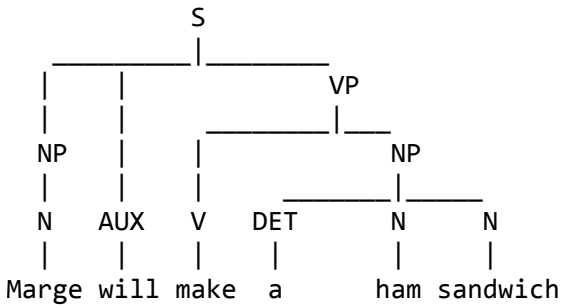
```
AUX
 |
will
```

**Excercise 2**

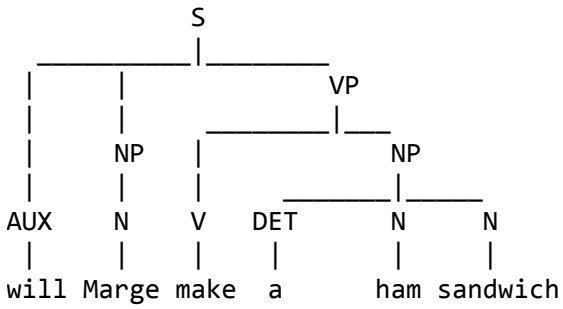
In [ ]:
```

```

**Exercise 3**

In [5]:
```
s1 = nltk.Tree.fromstring('(S (NP (N Marge)) (AUX will) (VP (V make) (NP (DET a)
s1.pretty_print()
```

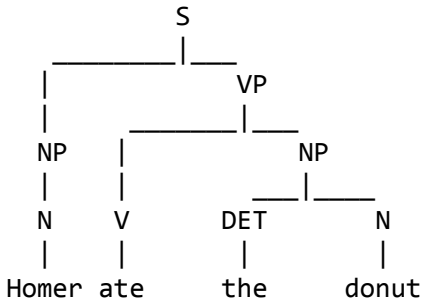



In [6]:
```
s2 = nltk.Tree.fromstring('(S (AUX will) (NP (N Marge)) (VP (V make) (NP (DET a)
s2.pretty_print()
```



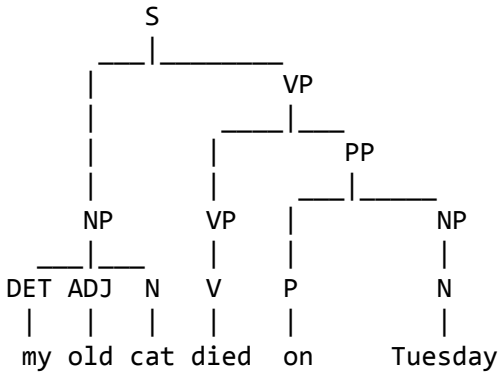

**Exercise 4**

In [7]:
```
s3 = nltk.Tree.fromstring('(S (NP (N Homer)) (VP (V ate) (NP (DET the) (N donut))
s3.pretty_print()
```

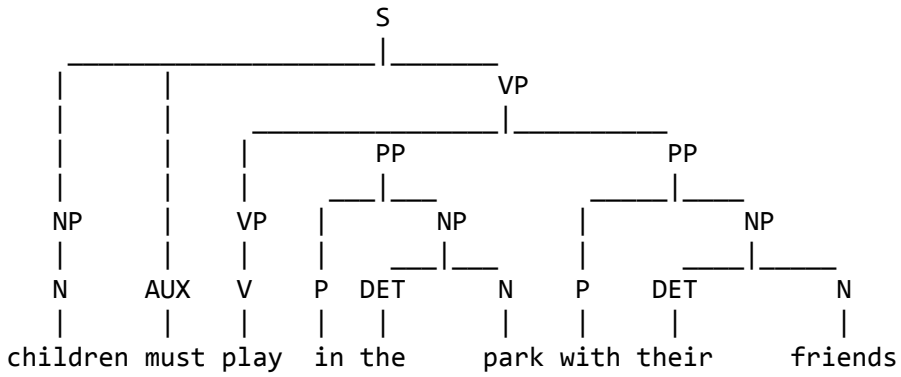



**Exercise 5**

In [8]:
```
s4 = nltk.Tree.fromstring('(S (NP (DET my)(ADJ old)(N cat))(VP(VP(V died))(PP(P o
s4.pretty_print()
```




In [9]:
```
s5 = nltk.Tree.fromstring('(S(NP (N children))(AUX must)(VP(VP(V play))(PP(P in)(
s5.pretty_print()
```




**Exercise 6**

In [10]:
```
print(vp) #this is from exercise 1
```

```
(VP (V make) (NP (DET a) (N ham) (N sandwich)))
```

In [11]:
```
vp_rules = vp.productions()
vp_rules
```

Out[11]:
```
[VP -> V NP,
 V -> 'make',
 NP -> DET N N,
 DET -> 'a',
 N -> 'ham',
 N -> 'sandwich']
```

In [12]:
```
vp_rules[0]
```

Out[12]:
```
VP -> V NP
```

In [13]:
```
vp_rules[1]
```

Out[13]:
```
V -> 'make'
```

In [14]:
```
vp_rules[0].is_lexical()
```

Out[14]:
```
False
```

In [15]:
```
vp_rules[0].is_lexical()
```

Out[15]:
```
False
```

***Explore the CF rules of s5***

In [16]:
```
print(s5)
```

```
(S
  (NP (N children))
  (AUX must)
  (VP
    (VP (V play))
    (PP (P in) (NP (DET the) (N park)))
    (PP (P with) (NP (DET their) (N friends)))))
```

In [17]:
```
s5_rules=s5.productions()
s5_rules
```

Out[17]:
```
[S -> NP AUX VP,
 NP -> N,
 N -> 'children',
 AUX -> 'must',
 VP -> VP PP PP,
 VP -> V,
 V -> 'play',
 PP -> P NP,
 P -> 'in',
 NP -> DET N,
 DET -> 'the',
 N -> 'park',
 PP -> P NP,
 P -> 'with',
 NP -> DET N,
 DET -> 'their',
 N -> 'friends']
```

a. How many CF rules are used in s5?

In [18]:
```
print("How Many CF values are used in s5   ",len(s5_rules))
```

```
How Many CF values are used in s5    17
```

b. How many unique CF rules are used in s5?

In [19]:
```
x = npt.array(s5_rules)
print("How Many unique CF rules are used in s5  ",len(npt.unique(x)))
```

```
How Many unique CF rules are used in s5   15
```

c. How many of them are lexical?

In [20]:
```
n=0
for x in s5_rules:
  if x.is_lexical():
    n = n+1
print("How many of them are lexical?  ",n)
```

```
How many of them are lexical?   9
```