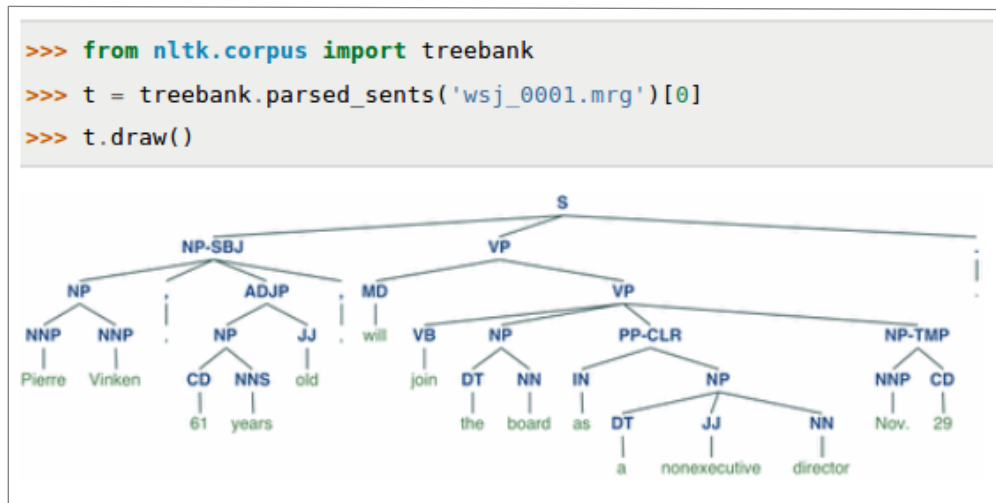


Gramáticas Libre de Contexto con Python y NLTK

NLTK - Natural Language Toolkit

- Librería de Python para procesamiento de lenguaje natural.
- Recursos, funcionalidades, documentación, comunidad, etc.
 - POS tagging
 - Parsers
 - Chunkers
 - Corpus
 - WordNet
 - Modelos Estadísticos



Imágen extraída de <https://www.nltk.org>

- <https://www.nltk.org>

NLTK - GLC

- Manejo de **gramáticas libre de contexto**
 - Reglas, gramáticas y árboles
 - Sintáxis para reglas, *parsers*, etc.

```
from nltk import CFG

grammar = CFG.fromstring("""
S -> SN SV
SN -> Det N
SV -> V | V N
Det -> 'el'
N -> 'perro' | 'gato' | 'huesos'
V -> 'duerme' | 'come'
""")
```

<Grammar with 10 productions>

```
grammar.start()
```

S

```
grammar.productions()
```

```
[S -> SN SV,
 SN -> Det N,
 SV -> V,
 SV -> V N,
 Det -> 'el',
 N -> 'perro',
 N -> 'gato',
 N -> 'huesos',
 V -> 'duerme',
 V -> 'come']
```

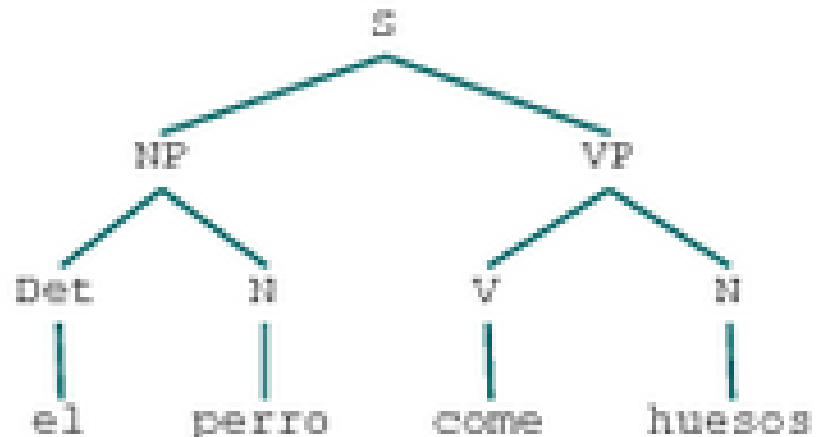
NLTK – GLC (Reconocer)

- Reconocer con una GLC (parse)

```
sent = ['el', 'perro', 'come', 'huesos']  
parser = nltk.ChartParser(grammar)  
  
for tree in parser.parse(sent):  
    print(tree)
```

```
(S (NP (Det el) (N perro)) (VP (V come) (N huesos)))
```

```
tree.draw()
```



NLTK – GLC (Generar)

- Generar tiras a partir de una GLC

```
from nltk.parse.generate import generate

for s in generate(grammar, n=8):
    print(' '.join(s))
```

```
el perro duerme
el perro come
el perro duerme perro
el perro duerme gato
el perro duerme huesos
el perro come perro
el perro come gato
el perro come huesos
```

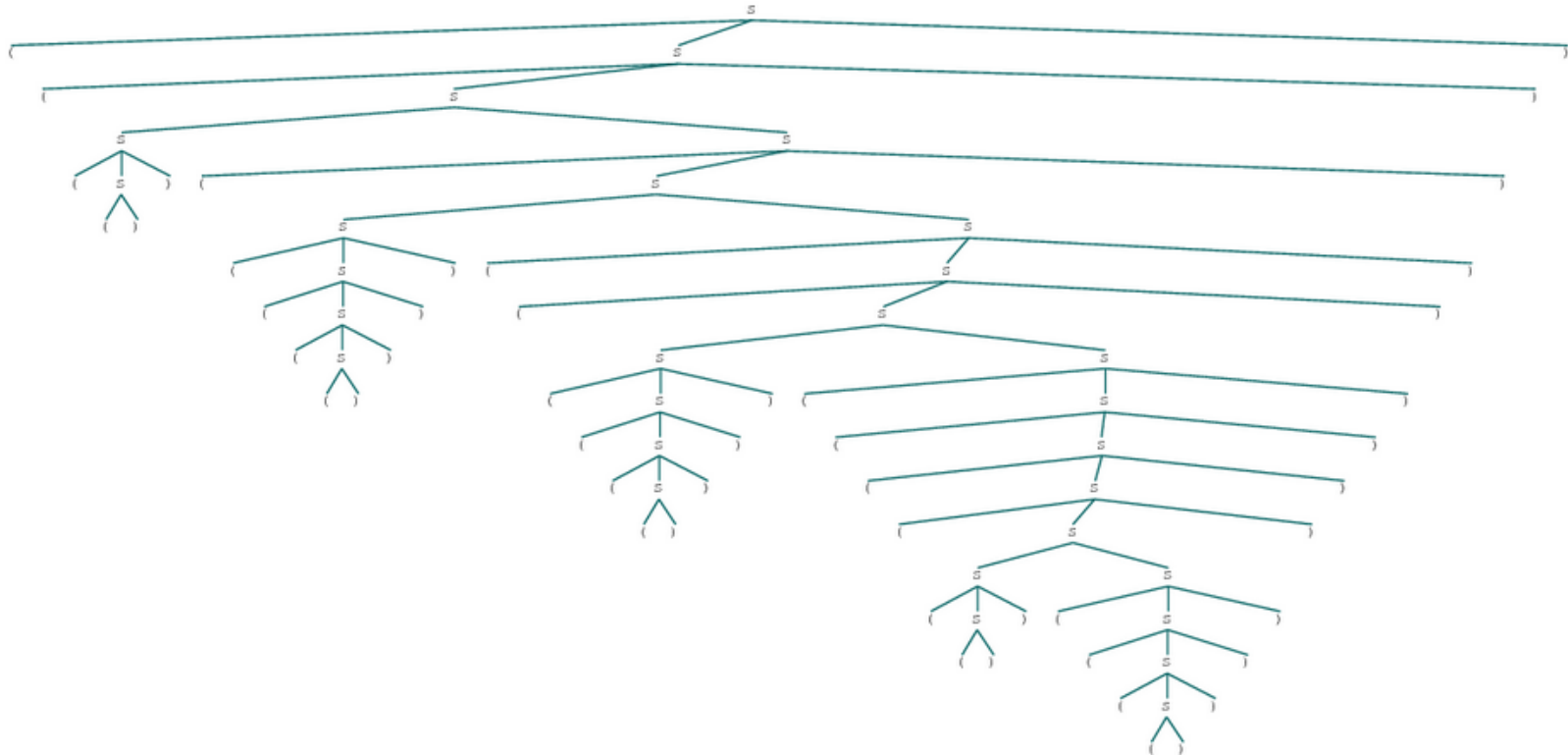
Ejemplo

- Tiras de paréntesis anidados.

```
from nltk import CFG

g = CFG.fromstring("""
S -> '(' ')' | '(' S ')' | S S
""")
```

Ejemplo

[illegible]

Ejemplo

```
sent2 = list('(((((((()))'))))')
len(list(parser.parse(sent2)))
```



```
sent3 = list('(((((((((((((())())))))))()())()()()()))')  
len(list(parser.parse(sent3)))
```

5

Tree

- La clase **`nltk.tree.Tree`**
 - Funciones
 - `label`
 - `flatten`
 - `subtrees`
 - `leaves`
 - `height`
 - ...
 - Iterar sobre los hijos

```
for node in t:
```

Tree

- Distinguir variables y terminales

```
from nltk.tree import Tree

for node in t:
    if type(node) is Tree:
        print('Variable: ', node.label())
    else:
        print('Terminal: ', node)
```

```
Terminal:  (
Variable:  S
Terminal:  )
```

Máximo Anidamiento

- Nivel máximo de anidamiento

```
from nltk.tree import Tree

def depth(t):
    d = 0
    for node in t:
        if type(node) is Tree:
            d = max(depth(node)+1, d)
        else:
            d = max(1, d)
    return d
```

Máximo Anidamiento

- Ejemplos de invocación

```
depth(list(parser.parse(list('((( )))')))[0])
```

3

```
depth(list(parser.parse(list('(()()()()())')))[0])
```

6

Máximo Anidamiento

- Distinguimos si la regla introduce paréntesis

```
def depth2(t):  
    d = 0  
    flag = False  
    for node in t:  
        if type(node) is Tree:  
            if flag:  
                d = max(depth2(node)+1, d)  
            else:  
                d = max(depth2(node), d)  
        else:  
            flag = True  
            d = max(1, d)  
    return d
```

Máximo Anidamiento

- Ejemplos de invocación

```
depth2(list(parser.parse(list('((( )))')))[0])
```

3

```
depth2(list(parser.parse(list('(()()()()())')))[0])
```

2