

Teoría de Lenguajes

Curso 2022

Laboratorio 2 – Gramática Libre de Contexto

Este laboratorio consiste en utilizar las funcionalidades de gramática libre de contexto (GLC) de la librería NLTK [1] para implementar un conjunto de programas en Python [2].

Cada programa recibe una entrada que es procesada con una GLC y despliega una salida. En cada programa se deberá construir una gramática para reconocer la entrada, basándose en la descripción del programa y un conjunto de ejemplos de entrada y salida. En algunos casos resultará conveniente construir una rutina que utilice el árbol de *parsing* de la entrada, en caso de haber alguno, para generar la salida.

Modo de trabajo

Cada programa a implementar es un módulo **Python 3** (versión 3.10) ejecutable que recibe como parámetros los nombres de los archivos de entrada y salida.

Todos los programas tienen lo siguiente en común:

- Analizan la entrada con una gramática libre de contexto
- En caso de que la entrada no pertenezca al lenguaje generado por la gramática el programa devuelve: **NO PERTENECE**
- En caso de que el conjunto de terminales de la gramática no cubra el conjunto de símbolos de la entrada el programa devuelve: **NO CUBRE**
- En caso de que la entrada pertenezca al lenguaje generado por la gramática el programa devuelve **PERTENECE** o una salida construida a partir del árbol de derivación de la entrada.

Se permite importar y reutilizar funciones entre los módulos pero no crear módulos auxiliares.

Para realizar este laboratorio se imparte el archivo **lab2.zip** que contiene ejemplos de entrada y salida, el módulo *programa0.py* con un programa de ejemplo implementado y el script (*ejecutar.py*) que ejecuta cada programa con las entradas disponibles y compara las salidas obtenidas con las esperadas.

programa0.py (implementado)

Este programa verifica que la tira de entrada pertenezca al lenguaje: $\{a^n b^n \text{ con } n > 0\}$

Este programa se encuentra implementado como ayuda para la implementación de los restantes programas.

La gramática que se entrega implementada (con la sintáxis de NLTK) es:

```
S -> 'a' S 'b' | 'a' 'b'
```

Ejemplo de entrada 1:

aabb

Ejemplo de salida 1:

PERTENECE

Ejemplo de entrada 2:

aab

Ejemplo de salida 2:

NO PERTENECE

Ejemplo de entrada 3:

aa11

Ejemplo de salida 3:

NO CUBRE

programa1.py

Este programa reconoce al lenguaje $\{ a^n b^m c^k : (n=m \vee m=k \vee n=k) \wedge \min(n,m,k) > 0 \}$

Ejemplo de entrada 1:

aaabccc

Ejemplo de salida 1:

PERTENECE

Ejemplo de entrada 2:

abbccc

Ejemplo de salida 2:

NO PERTENECE

Ejemplo de entrada 3:

aabbccdd

Ejemplo de salida 3:

NO CUBRE

programa2.py (lógica propopsicional)

Este programa reconoce fórmulas de la lógica proposicional (e.g. “p and not q”) y las reescribe en notación prefija (e.g. “and(p, not(q))”).

Las fórmulas están formadas por los siguientes elementos:

- operadores: not, and, or
- paréntesis: (,)
- letras de proposición: a,...,z

Puede asumir que no hay problemas de ambigüedad por el orden de los operadores en la entrada.

Todos los operadores y las letras de proposición están en minúsculas.

Ejemplo de entrada 1:

(not (a or b)) and c

Ejemplo de salida 1:

and(not(or(a,b)),c)

Ejemplo de entrada 2:

q

Ejemplo de salida 2:

q

Ejemplo de entrada 3:

not a b

Ejemplo de salida 3:

NO PERTENECE

Ejemplo de entrada 4:

no a

Ejemplo de salida 4:

NO CUBRE

programa3.py (lenguaje natural - voz pasiva)

Este programa recibe como entrada una oración de un conjunto que se detalla a continuación y la retorna en voz pasiva.

Las oraciones recibidas tienen la forma "GN V GN" donde:

- V es un verbo
- GN es un grupo nominal formado por un nombre propio o un determinante y un nominal.

Los grupos nominales son formados con los siguientes elementos:

Determinantes: la, el, un, una, esa, ese, tu

Nominales: manzana, banana, naranja, kiwi, pera, frutilla, mango, perro, perra, gato, gata, elefante, hueso, pescado

Nombres propios: Juan, Marta, Pedro, Julia

Los verbos considerados son los siguientes:

- come / es comida por / es comido por
- salta / es saltada por / es saltado por
- trepa / es trepada por / es trepado por
- descubre / es descubierta por / es descubierto por
- aplasta / es aplastada por / es aplastado por

Las oraciones deben de ser sintácticamente correctas, incluyendo la concordancia del género gramatical. Por ejemplo, se considera incorrecto el uso del determinante "la" junto al nominal "perro", para formar el grupo nominal "la perro". Análogamente se deberá controlar esta concordancia al generar la voz pasiva.

Únicamente los nombres propios tienen la primera letra en mayúscula. Las mayúsculas de principio de oración son omitidas.

No se consideran símbolos de puntuación ni ninguna otra palabra que las detalladas anteriormente.

Ejemplo de entrada 1:

el elefante aplasta una manzana

Ejemplo de salida 1:

una manzana es aplastada por el elefante

Ejemplo de entrada 2:

Pedro come un mango

Ejemplo de salida 2:

un mango es comido por Pedro

Ejemplo de entrada 3:

Juan descubre un frutilla

Ejemplo de salida 3:

NO PERTENECE

Ejemplo de entrada 4:

Juliana descubre un perro

Ejemplo de salida 4:

NO CUBRE

Entrega

La entrega es el **24 de junio a las 23:55** y se realizará mediante un formulario en EVA que estará disponible próximo a la fecha de entrega.

Los grupos deben ser los mismos que en el laboratorio 1.

Se debe entregar:

- los archivos **programa{1,2,3}.py** con los programas implementados
- un archivo **integrantes.txt** con las cédulas de los integrantes del grupo (una por línea) sin puntos ni dígito de verificación.

Referencias

[1] <https://www.nltk.org/book/ch08.html>

[2] <https://docs.python.org/3/>