

# Análisis sintáctica

(Consulta la fecha de entrega en el classroom del curso)

## Reglas:

- No está permitido utilizar códigos de otras personas o del Internet. Puede usar bibliotecas estándar para estructuras de datos basicas.
- El programa debe implementarse **individualmente**.
- Lenguajes permitidos: Java, C, C++, Lisp o Python.
- Entrega el **código fuente** en un solo archivo.
- Su programa se compilará en C99, C++11, SBCL o Java 7 (no inicie el archivo java con “package” como es el valor predeterminado de Netbeans). Para los programas de Python, informe la versión utilizada (2 o 3) y colóquela en la primera línea del script usando *shebang*.
- Los programas con errores de sintaxis serán descartados inmediatamente.
- El programa debe seguir estrictamente las especificaciones de entrada y salida, ya que serán automáticamente probadas por un *script*.
- ¡La fecha de entrega no se extenderá!

## 1. Objetivo

En este trabajo, implementará el algoritmo CYK (de Cocke, Kasami y Younger) para analizar sintácticamente un texto de acuerdo con una gramática dada en la forma normal de Chomsky. Este es el algoritmo *parsing* que usa programación dinámica (investigar).

## 2. Especificación da entrada

a entrada debe leerse desde **entrada estándar** y contendrá una descripción de una gramática en la forma normal de Chomsky. Los terminales serán todos los caracteres ASCII visibles excepto las letras mayúsculas. Más concretamente, los terminales son caracteres con códigos ASCII del 33 al 64 y del 91 al 126. Cada variable se representará con una letra mayúscula (caracteres con código del 65 al 90).

La primera línea de la entrada contiene un número  $n \leq 1000$  que será el número de reglas, y que debe ser positivo. Cada una de las siguientes líneas  $n$  describe una regla. El primer carácter en la línea de una regla es una variable. Si la línea contiene solo el carácter variable (sin contar el marcador de final de línea '\n'), la regla codifica una regla como  $\varepsilon$ . Si la línea tiene sólo dos caracteres, el primero es la variable y el segundo es una terminal. Si la línea tiene tres caracteres, los dos últimos también deben ser variables, ya que la gramática se dará en la forma normal de Chomsky. La variable inicial es la primera variable de la primera regla.

### 3. Especificación de la salida

Imprime, para cada línea después de la descripción de la gramática, una línea que contenga **SIM** si la gramática genera la palabra leída y **NAO** (sin tilde) en caso contrario.

### 4. Ejemplo

```
15
SDB
SFC
SAA
SDF
SFD
S
ADB
AFC
AAA
ADF
AFD
BAF
CAD
Da
Fb
abbb
aababb
```

En el ejemplo anterior, su programa debe decidir si las palabras **abbb** y

aababb son generadas por la gramática:

$$S \rightarrow DB \mid FC \mid AA \mid DF \mid FD \mid \varepsilon$$

$$A \rightarrow DB \mid FC \mid AA \mid DF \mid FD$$

$$B \rightarrow AF$$

$$C \rightarrow AD$$

$$D \rightarrow a$$

$$F \rightarrow b$$

La salida correcta del ejemplo es:

NAO  
SIM

## Agradecimientos

Agradecemos al Prof. Cristiane Maria Sato, quien concibió la primera versión de este ejercicio y al profesor Daniel Martins quien generó el script de auto corrección.