CYK

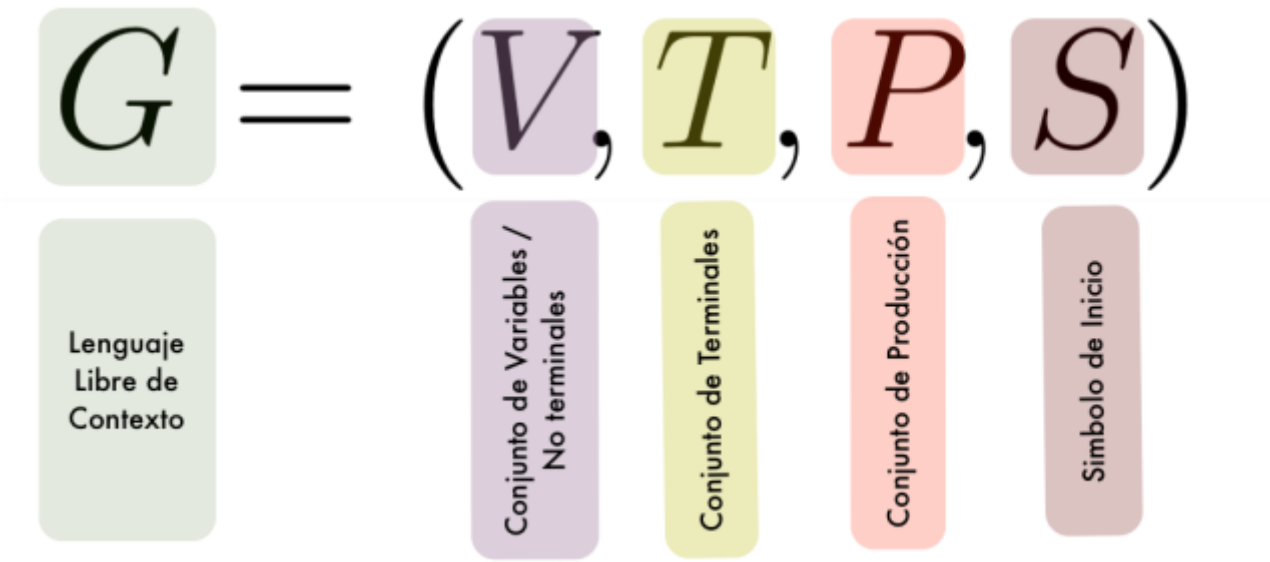
Daniel Velázquez Lara A01636246

Introducción y Propósito

El propósito es básicamente explicar el proyecto de segundo parcial de la materia de Matemáticas Computacionales,que es un **analizador sintáctico**, usando el algoritmo **CYK**. Además de que se va a mostar su funcionalidad, diseño e implementación. Además de que se van a explicar de manera muy breve los temas con los que se trabaja este proyecto, como lo es la **Forma Normal de Chomsky**(FnCH) y las **Gramáticas Libres de Contexto**(GLC)

Teoría

Gramáticas Libres de Contexto (GLC)



- **Definición Formal de GLC**

```
G = (N, Σ, P, S), donde:

• N es un conjunto finito, no vacío de símbolos no terminales
• Σ es el alfabeto de símbolos terminales
• S ∈ N es el símbolo inicial
• P es un conjunto de producciones
    – Estas producciones son de la forma A → α, donde A ∈ N, α ∈ (N ∪ Σ)*,
es decir α puede ser:
        – Una cudena de símbolos que contiene tanto símbolos no terminales
como terminales
        – Sólo símbolos no terminales o Sólo símbolos terminales
        – α=ε

Si G = (N, Σ, P, S) es una gramatica libre de contexto, entonces el
conjunto de todas las cadenas de símbolos terminales derivables a partir
```

de S es un lenguaje sobre Σ, al que llamamos el lenguaje generado por G.
$L(G)=\{w \in \Sigma_* | S \Rightarrow_* w\}$

- Ejemplos de la vida real que usan GLC
  - Lenguajes de programación
  - Inteligencia Artificial

Forma Normal de Chomsky

$$
\begin{aligned}
\$ &\longrightarrow W_bDD \mid CW_a \mid W_bW_c \\
A &\longrightarrow B \mid W_aCC \mid W_bW_aD \\
B &\longrightarrow W_cBD \mid \epsilon \mid AC \\
C &\longrightarrow W_bD \mid W_aBA \\
D &\longrightarrow CD \mid W_a \\
W_a &\longrightarrow a \\
W_b &\longrightarrow b \\
W_c &\longrightarrow c
\end{aligned}
$$

- Definición Formal de FNCh

Una gramática está en la Forma Normal de Chomsky si todas sus producciones son de alguna de las siguientes formas:
$A \rightarrow BC$
$A \rightarrow \propto$

Donde A, B y C forman parte del conjunto de símbolos no terminales y $\propto$ pertenece al conjunto de los símbolos terminales

Algoritmo CYK

- Algoritmo que determina si una cadena de texto pertenence a una gramática libre de contexto
  - Además de que muestra cómo puede ser generada (Análisis sintáctico)
  - Normalmente la GLC debe estar en la Forma Normal de Chomsky (FNCh), pero no es necesario para que este sea considerado un CYK

Diseño

Implementación

- Como correr el programa
  - Al programa se le deben de pasar 2 argumentos extras a la hora de correr el programa: el diccionario(GLC) y la frase (String)
    - La frase va a ser la palabra a la cual se le va a verificar si pertenece a la GLC o no

- El diccionario o gramática a la cual va a ser interpretado el programa

```
javac CYK.java
java CYK grammar.txt "frase"
```

- **Ejemplo**

```
→  CYK java CYK grammar.txt abbbabaa
Word: abbbabaa

G = ({a, b}, {S, A, B, E, C, X, Y, Z}, P, S)

Producciones(Derivaciones) P:
A -> a | YE | XC
B -> b | XE | YZ
C -> AA
E -> YB | XA
S -> YB | XA | *
X -> b
Y -> a
Z -> BB

Resultado CYK:

+-------+-------+-------+-------+-------+-------+-------+-------+
| a     | b     | b     | b     | a     | b     | a     | a     |
+-------+-------+-------+-------+-------+-------+-------+-------+
| A,Y   | B,X   | B,X   | B,X   | A,Y   | B,X   | A,Y   | A,Y   |
+-------+-------+-------+-------+-------+-------+-------+-------+
| E,S   | Z     | Z     | E,S   | E,S   | E,S   | C     |
+-------+-------+-------+-------+-------+-------+-------+
| B     | -     | B     | B     | A     | A     |
+-------+-------+-------+-------+-------+-------+
| Z     | Z     | Z     | E,S   | C     |
+-------+-------+-------+-------+-------+
| B     | -     | B     | A     |
+-------+-------+-------+-------+
| Z     | Z     | E,S   |
+-------+-------+-------+
| B     | B     |
+-------+-------+
| E,S   |
+-------+

La palabra abbbabaa SI es un elemento de la GLC G
```

- **Ejemplo GLC**

```
S
a b
S A B E C X Y Z
```

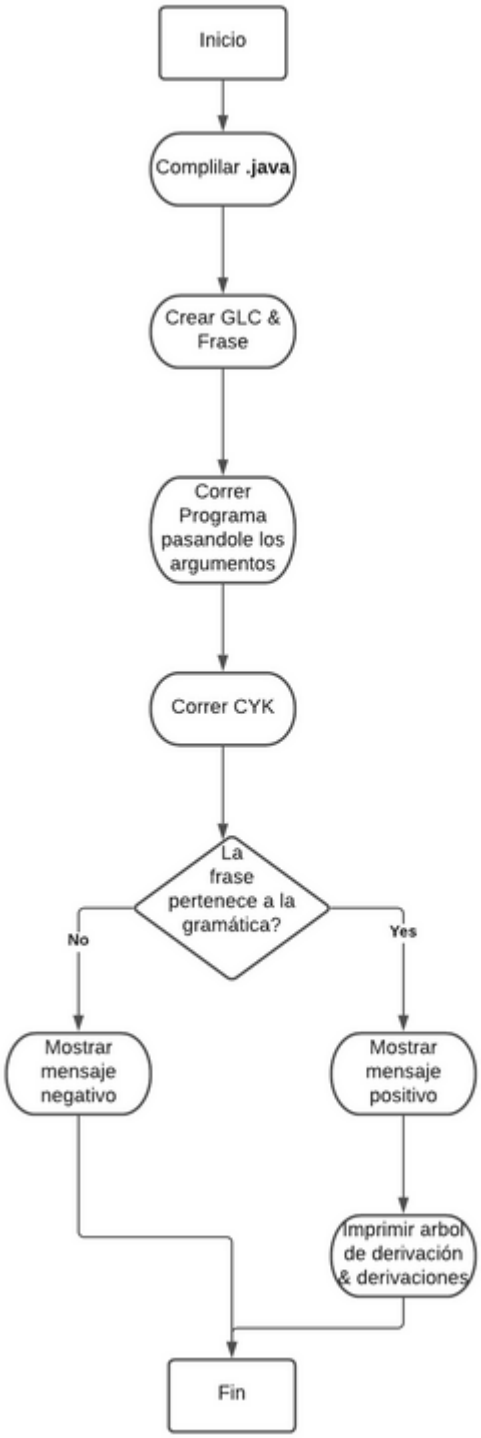
```
S YB XA *
E YB XA
A a YE XC
B b XE YZ
C AA
X b
Y a
Z BB
```

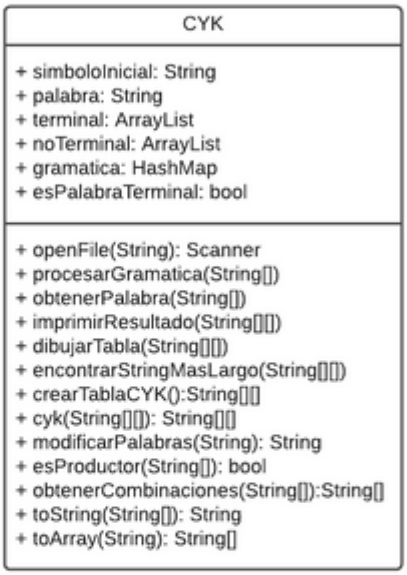
Funcionamiento

> *Cada parte del código se explica detalladamente en el video*

- openFile(String): Scanner
  - Abrir archivo de texto que contiene la gramática
- procesarGramatica(String[])
  - Usa la clase Scanner para abrir el archivo y procesa cada símbolo para poder tener un análisis sintáctico adecuado, utlizando Hashmaps para su correcta busqueda
- obtenerPalabra(String[])
  - Checa si la frase es un símbolo terminal o no
- imprimirResultado(String[][])
  - imprime la tabla resultante del algoritmo CYK, mostrando la frase según la GLC
- dibujarTabla(String[][])
  - Se procesa cada símbolo, en combinación con la tabla para poder almacencar de manera correcta los resultados. Se va a iterar cada parte del array para ir llenandolo con los valores correctos.
- crearTablaCYK():String[][]
  - Se crea una tabla para poder indexar de mancera correcta los resultados del algoritmo CYK, en base a la GLC
- cyk(String[][]): String[][]
  - La parte más importante del programa, en donde se hace uso de de un array bidimensional para llenarlos con símbolos no terminales de la frase dada. Este hace uso de la programación dinámica para poder llegar a la solución de maner eficaz y práctica.
- esProductor(String[]): bool
  - Se checa en el hashmap si la palabra pasa el análisis sintáctico para poder ser agregado o no.
- toString(String[]): String
  - Imprimir el objecto CYK en texto
- toArray(String): String[]
  - Convertir la frase dada en un array para poder usarse más adelante

Diagrama de clases y de flujo





Referencias

- Forma Normal de Chomsky
- Gramáticas Libres de Contexto
- Algoritmo CYK