Practica Modelos 1

Algoritmo de Decisión -- CYK

Alejandro Lillo Rodriguez

https://github.com/AlejandroLilloRodriguez/PracticaModelos.git

Introducción

El objetivo de este trabajo es llevar a cabo la implementación del algoritmo CYK, explorando su funcionalidad y cómo puede aplicarse a la hora de comprobar si una palabra pertenece o no a la gramática . A través de esta implementación, buscamos comprender el funcionamiento del algoritmo CYK y su capacidad para descomponer una cadena de palabras en . Su funcionamiento se basa en la construcción de una tabla bidimensional y la aplicación de reglas de producción para verificar la existencia de derivaciones en la gramática.

El algoritmo CYK opera de la siguiente manera:

1. Se parte de una gramática libre de contexto en forma normal de Chomsky, que define las reglas de producción y los símbolos no terminales y terminales.

- La cadena de entrada se divide en componentes básicos, generalmente caracteres individuales o símbolos terminales.

2. Se crea una tabla bidimensional (tabla CYK) con filas representando los símbolos no terminales y columnas representando las posiciones en la cadena de entrada.

- Se llenan las celdas de la tabla CYK utilizando las reglas de producción de la gramática. Se comparan las combinaciones de símbolos terminales para determinar si se pueden derivar símbolos no terminales.

- El proceso de llenado se realiza de manera recursiva, combinando las celdas adyacentes y aplicando las reglas de producción hasta llegar a la celda correspondiente al símbolo inicial de la gramática.

3. Si el símbolo inicial está presente en la celda correspondiente a la posición inicial de la cadena de entrada, se considera que la cadena pertenece a la gramática.

4. Si no se encuentra el símbolo inicial en la celda correspondiente, la cadena no pertenece a la gramática.

2. Decisiones de diseño tomadas en la práctica y cómo se han implementado.

En la implementación del algoritmo CYK, he utilizado varias estructuras de datos para organizar los diferentes elementos del algoritmo, como por ejemplo: elementos terminales, no terminales, axioma y producciones.
Estas estructuras de datos son:

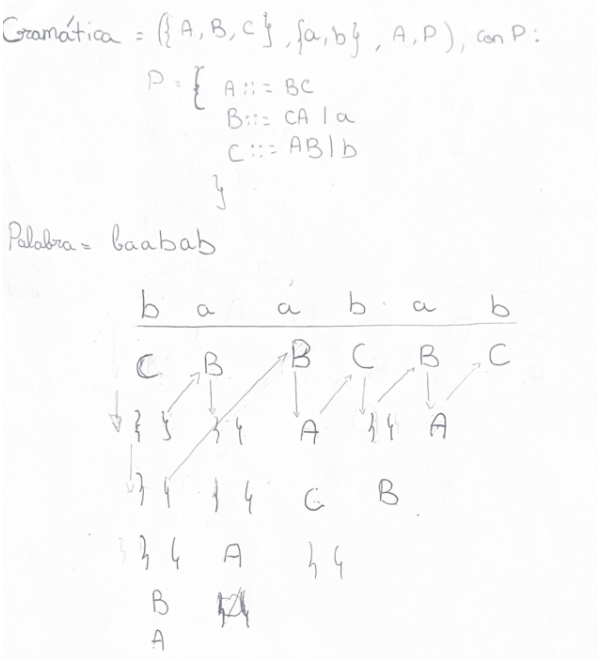
HashMap: Donde hago referencia al conjunto de producciones ya que al tener dos parámetros es mas facil implementarlas en este conjunto.Se utiliza para almacenar las producciones de cada elemento no terminal. La clave del mapa es un carácter que representa un elemento no terminal, y el valor es un conjunto de cadenas que representan las producciones asociadas a ese elemento no terminal.

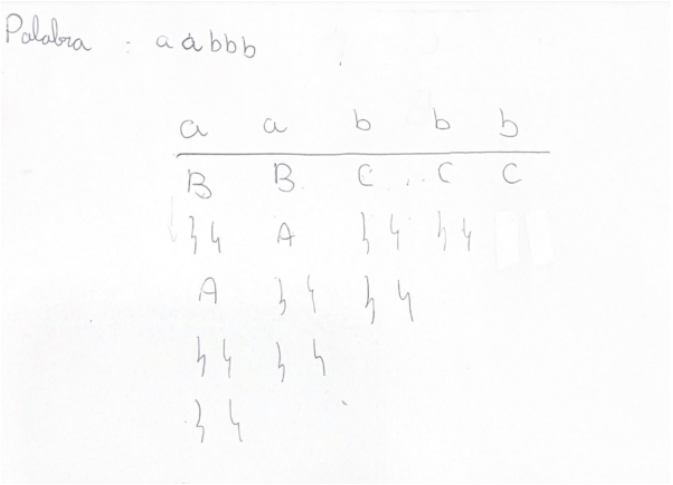
HashSet: Se utilizan dos conjuntos, noTerminales y terminales, para almacenar los elementos no terminales y terminales respectivamente. Estos conjuntos se utilizan para verificar la validez de los elementos no terminales y terminales introducidos, así como para realizar búsquedas rápidas.

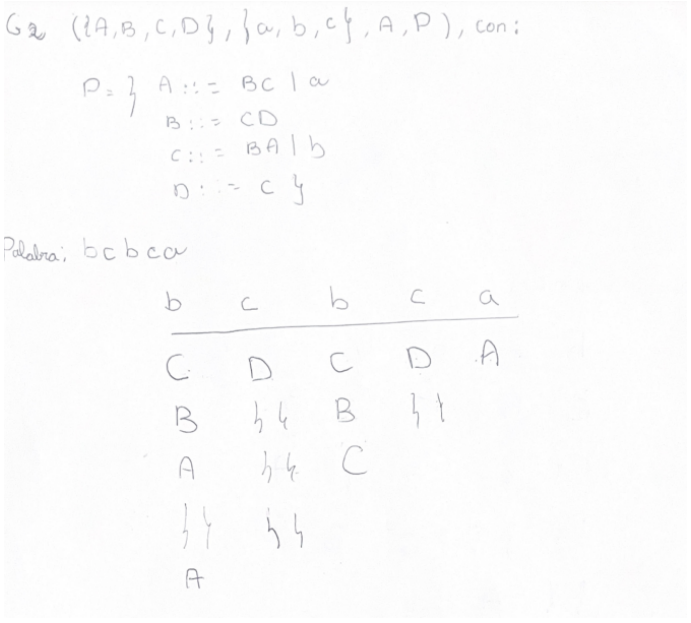
ArrayList[][]: La matriz llamada "tabla" se utiliza para representar la tabla utilizada en el algoritmo CYK. Esta matriz se inicializa con un tamaño basado en la longitud de la palabra que se está analizando. Cada celda de la matriz es un conjunto de caracteres no terminales que representan los posibles elementos no terminales que pueden derivar en esa posición de la palabra.

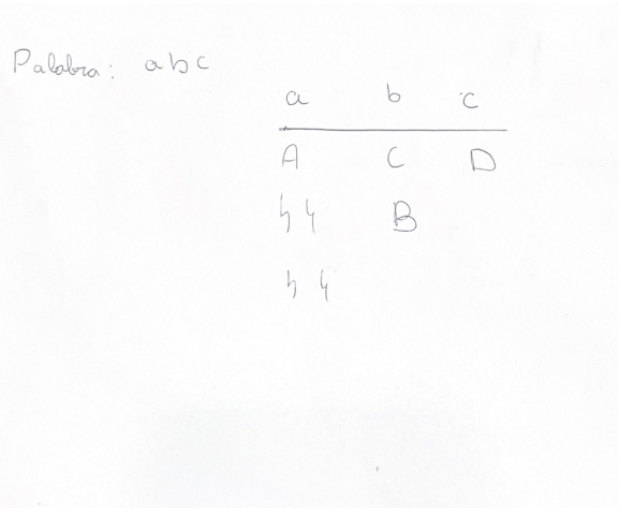
Estas estructuras de datos son útiles para usarlas en los métodos para agregar elementos no terminales y terminales, establecer el axioma, agregar producciones y obtener información sobre la gramática.

3. Descripción de cada una de las pruebas de tests que el alumno ha definido.









P = { S ::= AB
A ::= BS | a
B ::= SA | b | DC
C ::= a
D ::= b }

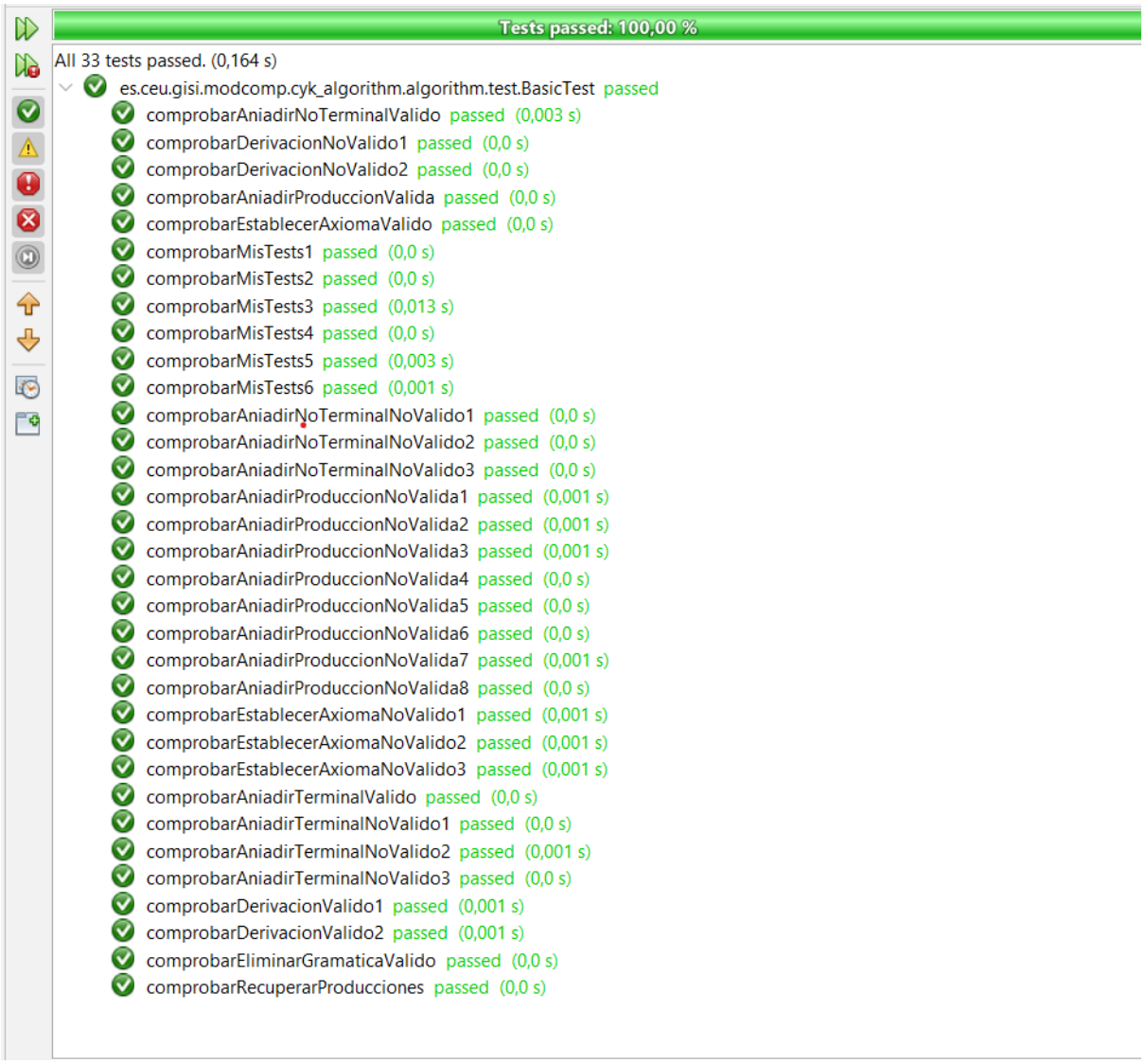
Palabra: bababa

b	a	b	a	b	a
BD	AC	BD	AC	BD	AC
B	S	B	S	B	
A	SB	A	SB		
A	{ }	A			
S	{ }				
S, B					

Palabra: aaaaba

a	a	a	a	b	a
AC	AC	AC	AC	BD	AC
{ }	{ }	{ }	{ }	{ }	{ S }
{ }	{ }	{ }	{ }	{ }	
{ }	{ }	{ }	{ }		
{ }	{ }				
{ }					

4.Completitud de los tests definidos en el proyecto y los definidos por el alumno



Conclusiones

En resumen, la implementación del algoritmo CYK presentada proporciona una estructura de código eficiente y organizada para realizar el análisis sintáctico de una gramática dada. Este código puede ser utilizado como una alternativa a la hora de determinar la pertenencia de una palabra al lenguaje generado por una gramática dada y no tener que recurrir a alternativas externas. También me ha ayudado a comprender cómo trabaja el algoritmo CYK internamente para poder saber si la palabra pertenece o no al lenguaje.
De forma general, calculo que habré estado en torno a 35 horas tratando de entender el algoritmo y aparte implementando mediante código la práctica.
Durante el trabajo del algoritmo, me fueron muy útiles la clase de introducción a la práctica y las tutorías de dudas ya que mediante esto las ideas me quedaron más claras para posteriormente hacer de forma más sencilla el código