Algoritmo CYK

El algoritmo de Cocke-Younger-Kasami (CYK) determina si una cadena puede ser generada por una gramática libre de contexto y, si es posible, cómo puede ser generada. Este proceso es conocido como análisis sintáctico de la cadena. El algoritmo es un ejemplo de programación dinámica.

La versión estándar de CYK reconoce lenguajes definidos por una gramática libre de contexto escrita en la forma normal de Chomsky (CNF). Cualquier gramática libre de contexto puede ser convertida a CNF sin mucha dificultad, CYK puede usarse para reconocer cualquier lenguaje libre de contexto. Es posible extender el algoritmo CYK para que trabaje sobre algunas gramáticas libre de contexto no escritas como CNF. Esto puede hacerse para mejorar la ejecución, aunque hace el algoritmo más difícil de entender.

Usaremos el programa para probar el siguiente caso.

Ejemplo

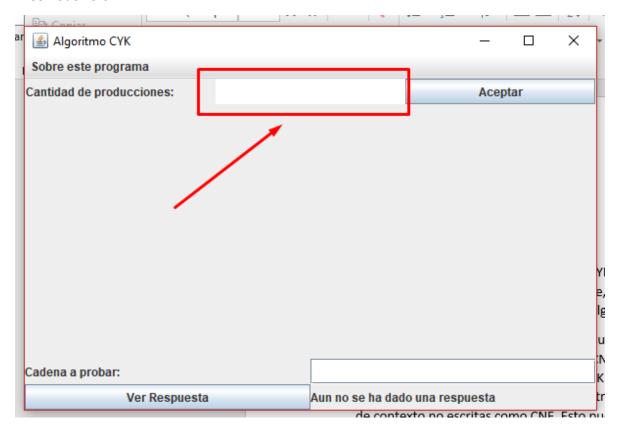
Vamos a aplicar el algoritmo CYK a la gramática:

$$G: \begin{cases} S \to BA \mid AC \\ A \to CC \mid b \\ B \to AB \mid a \\ C \to BA \mid a \end{cases}$$

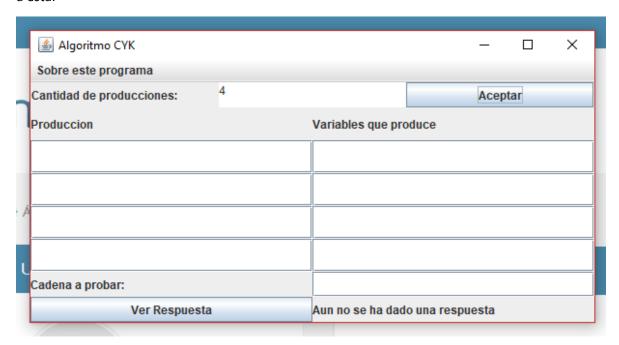
y a la cadena w = bbab. Se trata de determinar si $w \in L(G)$ o no. La tabla obtenida al hallar los X_{ij} , $1 \le i \le 4$, es la siguiente:

¿Cómo funciona este programa?

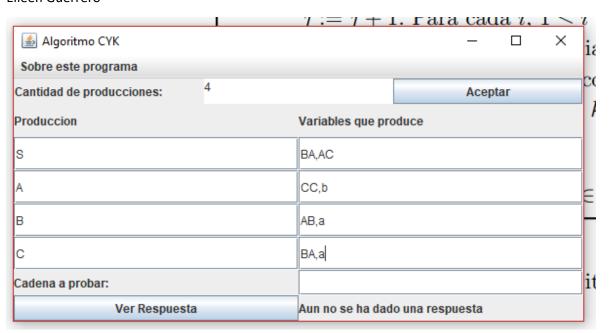
Al iniciar el programa, primero **debes introducir la cantidad de producciones a evaluar** en el algoritmo CYK.



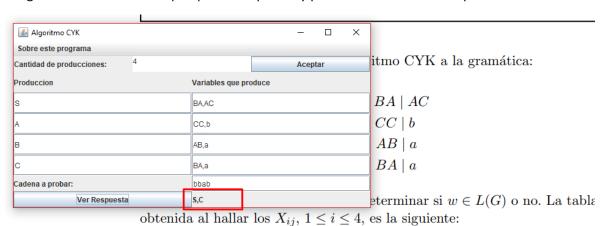
Para nuestro caso de ejemplo, que tenemos cuatro producciones, se mostrara una pantalla similar a esta.



Ahora se procede a llenar los datos. En la parte izquierda se escriben las producciones y en la parte derecha se escriben las variables que producen, separando cada variable con una coma como se muestra en la imagen.



Luego se escribe la cadena que queremos probar y presionamos en ver la respuesta.



j = 1j=2j = 3j=4bi = 1 $\{A\}$ $\{B\}$ $\{S,C\}$ i = 2 $\{A\}$ $\{S,C\}$ b $\{B,S\}$ i = 3 $\{B,C\}$ $\{S,C\}$ i = 1 $\{A\}$