

Complejidad CYK

El algoritmo CYK (Cocke-Younger-Kasami) es un algoritmo para determinar si una cadena pertenece al lenguaje generado por una gramática libre de contexto que está en forma normal de Chomsky (FNC). Este algoritmo es fundamental en el ámbito de la teoría de lenguajes formales y en el análisis sintáctico, particularmente en la validación de lenguajes formales como parte de compiladores.

Descripción del Algoritmo CYK

Dado una gramática libre de contexto en forma normal de Chomsky y una cadena de entrada de longitud n , el algoritmo trabaja construyendo una tabla bidimensional que permite verificar si la cadena puede ser generada por la gramática.

Pasos del algoritmo CYK:

1. **Conversión de la gramática a FNC:** El primer paso es asegurarse de que la gramática esté en **Forma Normal de Chomsky**. En esta forma, todas las reglas de producción son de la forma:
 - $A \rightarrow BCA \rightarrow BC$

- $A \rightarrow BC$ (donde A, B y C son variables o no terminales).
- $A \rightarrow a$ (donde A es un no terminal y a es un terminal).

2. **Inicialización:** Se construye una tabla $T[i][j]$ de tamaño $n \times n$, donde n es la longitud de la cadena de entrada. El índice i indica la posición inicial en la cadena y j indica la longitud del subsegmento que estamos analizando.

3. **Rellenar la tabla:** En la primera fase, se llena la diagonal de la tabla: para cada terminal en la cadena, se coloca el no terminal que lo produce según la gramática. En las siguientes fases, se usa la propiedad de las producciones de tipo $A \rightarrow BC$ para rellenar los elementos superiores de la tabla. Si una subsecuencia de la cadena de longitud j puede ser generada por la regla $A \rightarrow BC$, y las subsecuencias correspondientes a B y C están presentes en la tabla, se coloca A en la celda correspondiente de la tabla.

4. **Verificación:** Después de llenar la tabla, si el símbolo inicial S (de la gramática) está en la celda superior derecha de la tabla $T[1][n]$, entonces la cadena pertenece al lenguaje de la gramática. Si no, la cadena no es parte del lenguaje.

Complejidad del Algoritmo CYK

El análisis detallado de la complejidad del algoritmo se basa en tres factores principales:

Tamaño de la tabla: La tabla tiene $n(n+1)/2 \approx O(n^2)$ entradas debido a la forma triangular de la tabla que registra subsecuencias de diferentes longitudes.

Número de divisiones: Para una subsecuencia de longitud j , hay $j-1$ formas posibles de dividir la subsecuencia en dos partes no vacías. Por lo tanto, para cada celda, se realizan $O(j)$ operaciones, y j puede ser tan grande como n .

Número de comparaciones por producción: Para cada celda $T[i][j]$, buscamos todas las posibles combinaciones de no terminales que se pueden generar utilizando las reglas de producción de la forma $A \rightarrow BC$. Este proceso es de orden constante $O(1)$ para cada división.

En total, el costo es:

$$O(n^2) \times O(n) = O(n^3)$$

Esto demuestra que la complejidad temporal del algoritmo CYK es cúbica con respecto a la longitud de la cadena.

El algoritmo CYK es eficiente para analizar cadenas en lenguajes definidos por gramáticas libres de contexto en forma normal de Chomsky, con una complejidad de $O(n^3)$, lo que lo hace adecuado para cadenas de longitud moderada en aplicaciones como el análisis sintáctico de lenguajes de programación.