Algoritmo Cocke-Kasami-Younger

Autómatas y lenguajes formales

8 de octubre de 2023

Algoritmo Cocke-Kasami-Younger

Iniciamos con una gramática libre de contexto (evito un poco las siglas, que harían más tratable el texto, pero también apoyan a confundirse y pensar que se lee algo complicadísimo), que es un cuarteto como hemos definido con anterioridad:

$$G = \langle \Sigma, \Gamma, S, \to \rangle,$$

con

- Σ el alfabeto de la cadena de entrada (los símbolos terminales)
- Γ el alfabeto de las variables y terminales (es una combinación)
- S la variable inicial
- $\to$ el conjunto de las producciones.

La gramática son las reglas para generar cadenas en el lenguaje $L(G)$, pero ahora es la pregunta inversa ¿si yo les doy una cadena w cómo saben que es parte de ese lenguaje, $w \in L(G)$? En clase y en los ejercicios lo han hecho a pie, pero como nos gusta usar la computadora, y si la cadena tiene miles de caracteres, lo mejor es hacerlo por medio de un algoritmo.

Este algoritmo hecho por estos tres alegres compadres requiere que pongan su gramática en forma normal de Chomsky (vean el pdf que les mandé de ayuda, el que incluye el disco punk de Chomsky), esto es por la facilidad de hacer un árbol de acomodo con esta forma un poco más compacta.

Una notación útil antes de ver el algoritmo es el que caracteriza a las subcadenas de la cadena original $\alpha \in L$ con longitud n. Sea $0 \leq i < j \leq n$, con lo que:

$$\alpha_{i,j}$$

es la subcadena de α que va de la posición $i+1$ a la j. $T_{i,j} \subseteq \Gamma$ el conjunto de variables (o no terminales) que generan la subcadena $\alpha_{i,j}$. Como pueden ver el algoritmo irá comprobando por cortes si la cadena pertenece al lenguaje.

El algoritmo puesto en pseudo código se ve más o menos así:

```
for i := 0 to n − 1 do
  T_{i,i+1} := ∅
  for all A → a ∈ G do
```

```
        if  a = α_{i,i+1}  then
          T_{i,i+1} := T_{i,i+1} ∪ {A}
for  m := 2  to  n  do
  for  i := 0  to  n − m  do
    T_{i,i+m} := ∅
    for  j := i + 1  to  i + m − 1  do
      for  A → BC ∈ G  do
        if  B ∈ T_{i,j} ∧ C ∈ T_{j,i+m}  then
          T_{i,i+m} := T_{i,i+m} ∪ A
```

No pongo todos los detalles como comentario pues no me alcanza el espacio, pero les explico con un ejemplo.

Imaginemos que ya tenemos nuestra gramática en la forma normal de Chomsky (tomo prestado el ejemplo de **??**):

$$S->AB|BC$$
$$A->BA|a$$
$$B->CC|b$$
$$C->AB|a$$

A partir de esto se construye una tabla triangular donde cada hilera corresponde a la longitud de la subcadena (vamos a checar que $w \in L(G)$ por partes).

				$\{A, C\}$
			$\{B\}$	
		$\{A, C\}$		
	$\{A, C\}$			
$\{B\}$				
b	a	a	b	a

Cuadro 1: Acomodo de la cadena en la tabla a llenar y primeros $T_{i,i+1}$ llenos.

Les doy la cadena: *baaba* de longitud 5. Armamos la tabla triangular como se muestra en la figura **??**. La palabra va acomodada en el orden, los valores que se llenan primero en el algoritmo, los $T_{i,i+1}$ son los de la diagonal invertida, como se muestra en la misma tabla. Esos valores corresponden a los no terminales que producen el terminal correspondiente, como podemos ver de la gramática la única producción que lleva al terminal b es $B \to b$. Por otro lado, para obtener a tenemos dos opciones: $A \to a$ y $C \to a$.

El siguiente paso, una vez teniendo esto es ver por pares (como si estuviéramos viendo un árbol binario, ahora veremos el nodo padre de cada dos, yendo de izquierda a derecha). Los primeros dos cuadros de la hilera inferior los desarrollamos como si hiciéramos un producto, siempre por pares pues es la forma normal de Chomsky:

$$\{B\}\{A, C\} = \{BA, BC\}$$

¿Qué variables producen este par de variables? Vemos de nuevo nuestras producciones de la gramática, vemos que $S \to BC$ y $A \to BA$, entonces en el segundo cuadro de la última hilera ponemos este par de variables $\{S, A\}$. Este es el término $T_{0,2}$

Las siguientes parejas (recuerden mantener el orden, no es lo mismo AC que CA):

$$\{A, C\} \wedge \{A, C\} = \{AA, AC, CA, CC\}$$

La única combinación que es posible es $B \to CC$, entonces el tercer cuadro de la penúltima hilera se escribe $\{B\}$. Esta es la entrada $T_{1,3}$.

Luego va $\{A, C\}\{B\} \implies \{AB, CB\}$ (el orden, acuérdense, el orden). ¿Que producciones generan estas combinaciones de variables? Hay dos que producen una misma combinación $S \to AB$ y $C \to AB$, entonces en el cuarto cuadro de la antepenúltima hilera se pone $\{S, C\}$, que es la entrada $T_{2,4}$

Por último para esa semi-diagonal vemos que $\{B\}\{AC\} \implies \{BA, BC\}$, tenemos $S \to BC$ y $A \to BA$, entonces en el último cuadro de la segunda hilera se pone $\{S, A\}$. La tabla va como se ve en la figura **??**.

				$\{A, C\}$
			$\{B\}$	$\{S, A\}$
		$\{A, C\}$	$\{S, C\}$	
	$\{A, C\}$	$\{B\}$		
$\{B\}$	$\{S, A\}$			
b	a	a	b	a

Cuadro 2: Avance de la tabla del algoritmo CKY.

Ahora, en los que siguen es donde hay que tener más cuidado, ya no es tan directa la regla de combinar las producciones de la izquierda y arriba, pues como la tabla tiene más entradas debemos tomar en cuenta las demás. Para facilitar el mostrarles como funciona el algoritmo numeraré la tabla triangular de la manera que se muestre en la figura **??**.

				$T_{4,5}$
			$T_{3,4}$	$T_{3,5}$
		$T_{2,3}$	$T_{2,4}$	$T_{2,5}$
	$T_{1,2}$	$T_{1,3}$	$T_{1,4}$	$T_{1,5}$
$T_{0,1}$	$T_{0,2}$	$T_{0,3}$	$T_{0,4}$	$T_{0,5}$
b	a	a	b	a

Cuadro 3: Nomenclatura de los miembros de la tabla.

Nos paramos en el cuadro $T_{0,3}$, la regla para las comparaciones es como sigue:

$$\begin{aligned} T_{i,j} = T_{0,3} \overset{compara}{\Longrightarrow} & (T_{i,i+1} \wedge T_{i+1,i+m}) \cup (T_{i,i+2} \wedge T_{i+2,i+m}) \\ & = (T_{0,1} \wedge T_{1,3}) \cup (T_{0,2}, T_{2,3}) \\ & = \{B\} \wedge \{B\} \cup \{S, A\} \wedge \{A, C\} \\ & = \{BB, SA, SC, AA, AC\} \end{aligned}$$

Esto quiere decir que ahora la producción más alejada a la izquierda con la más cercana arriba, y la producción más arriba con la contigua a la izquierda. ¿Alguna variable produce esas combinaciones de variables? Ninguna, entonces dejan vacía la casilla (o la marcan con símbolo vacío).

El siguiente cuadro sería el $T_{1,4}$:

$$\begin{aligned} T_{i,j} = T_{1,4} \overset{compara}{\Longrightarrow} & (T_{i,i+1} \wedge T_{i+1,i+m}) \cup (T_{i,i+2} \wedge T_{i+2,i+m}) \\ & = (T_{1,2} \wedge T_{2,4}) \cup (T_{1,3} \wedge T_{3,4}) \\ & = \{A, C\}\{S, C\} \cup \{B\}\{B\} \\ & = \{AS, AC, CS, CC, BB\} \end{aligned}$$

¿Alguna variable genera una de esas combinaciones? Ahora sí, el $B \to CC$, entonces en ese cuadro se pone $\{B\}$, es decir $T_{1,4} = \{B\}$.

El que sigue es $T_{2,5}$

$$\begin{aligned} T_{i,j} = T_{2,5} \overset{compara}{\Longrightarrow} & (T_{i,i+1} \wedge T_{i+1,i+m}) \cup (T_{i,i+2} \wedge T_{i+2,i+m}) \\ & = (T_{2,3} \wedge T_{3,5}) \cup (T_{2,4} \wedge T_{4,5}) \\ & = \{A, C\}\{S, A\} \cup \{S, C\}\{A, C\} \\ & = \{AS, AA, CS, CA, SA, SC, CA, CC\} \end{aligned}$$

De nueva cuenta la única combinación posible es CC que viene de $B \to CC$, entonces $T_{2,5} = \{B\}$. La tabla va como se muestre en la figura **??**.

				$\{A, C\}$
			$\{B\}$	$\{S, A\}$
		$\{A, C\}$	$\{S, C\}$	$\{B\}$
	$\{A, C\}$	$\{B\}$	$\{B\}$	
$\{B\}$	$\{S, A\}$	$\emptyset$		
b	a	a	b	a

Cuadro 4: Avance de la tabla.

El cuadro que sigue es el $T_{0,4}$:

$$\begin{aligned} T_{i,j} = T_{0,4} \overset{compara}{\Longrightarrow} & (T_{i,i+1} \wedge T_{i+1,i+m}) \cup (T_{i,i+2} \wedge T_{i+2,i+m}) \cup (T_{i,i+3} \wedge T_{i+3,i+m}) \\ & = (T_{0,1} \wedge T_{1,4}) \cup (T_{0,2} \wedge T_{2,4}) \cup (T_{0,3}, X_{3,4}) \\ & = \{B\}\{B\} \cup \{S, A\}\{S, C\} \cup \{\varnothing\}\{B\} \\ & = \{BB, SS, SC, AS, AC\} \end{aligned}$$

No hay producción que genere esas combinaciones, entonces $T_{0,4} = \varnothing$, la que sigue sería $T_{1,5}$:

$$\begin{aligned} T_{i,j} = X_{1,5} \overset{compara}{\Longrightarrow} & (T_{i,i+1} \wedge T_{i+1,i+m}) \cup (T_{i,i+2} \wedge T_{i+2,i+m}) \cup (T_{i,i+3} \wedge T_{i+3,i+m}) \\ =& (T_{1,2} \wedge T_{2,5}) \cup (T_{1,3} \wedge T_{3,5}) \cup (T_{1,4} \wedge T_{4,5}) \\ =& \{A, C\}\{B\} \cup \{B\}\{S, A\} \cup \{B\}\{A, C\} \\ =& \{AB, CB, BS, BA, BA, BC\} \end{aligned}$$

Aquí sí hay AB dos veces y BA: $S \to AB$, $C \to AB$ y $A \to BA$, es decir $X_{2,5} = \{S, C, A\}$.

Ahora vamos al jefe final:

$$\begin{aligned} T_{i,j} = T_{0,5} \overset{compara}{\Longrightarrow} & (T_{i,i+1} \wedge T_{i+1,i+m}) \cup (T_{i,i+2} \wedge T_{i+2,i+m}) \cup (T_{i,i+3} \wedge T_{i+3,i+m}) \cup (T_{i,i+4} \wedge T_{i+4,i+m}) \\ =& (T_{0,1} \wedge T_{1,5}) \cup (T_{0,2} \wedge T_{2,5}) \cup (T_{0,3} \wedge T_{3,5}) \cup (T_{0,4} \wedge T_{4,5}) \\ =& \{B\}\{S, C, A\} \cup \{S, A\}\{B\} \cup \{\varnothing\}\{S, A\} \cup \{\varnothing\}\{A, C\} \\ =& \{BS, BC, BA, SB, AB\} \end{aligned}$$

Tres de esas producciones son posibles $S \to AB|BC$, $A \to BA$ y $C \to AB$, por lo tanto $T_{0,5} = \{S, A, C\}$.

Se ha completado la tabla, y espero hayan visto el algoritmo. Como el cuadro $T_{0,5}$ no es vacío e incluye S, entonces la cadena está en el lenguaje. **Fin**.

				$\{A, C\}$
			$\{B\}$	$\{S, A\}$
		$\{A, C\}$	$\{S, C\}$	$\{B\}$
	$\{A, C\}$	$\{B\}$	$\{B\}$	$\{S, C, A\}$
$\{B\}$	$\{S, A\}$	$\emptyset$	$\emptyset$	$\{S, A, C\}$
b	a	a	b	a

Cuadro 5: La tabla final, si hay S en el cuadro inferior de la derecha, se acepta la cadena.

Y ahora `https://www.youtube.com/watch?v=oQHivjjOcnE`, para sacar el estrés.

Referencias

[1] Kozen, Dexter C. "Automata and Computability" Springer (1997)

[2] `https://web.cs.ucdavis.edu/~rogaway/classes/120/winter12/CYK.pdf`

[3] `https://courses.cs.washington.edu/courses/cse431/15sp/CKY.pdf`