Autómatas y Lenguajes Formales
Nota 09. Algoritmo CKY*

Noé Salomón Hernández S.

1. Algoritmo CKY

Este algoritmo lleva las iniciales de sus descubridores Cocke, Younger y Kasami. Recibe como entrada una gramática libre de contexto G y una cadena $x = x_1 \dots x_n$, y determina si la cadena forma parte del lenguaje generado por dicha gramática. Suponemos además que la gramática G está en forma normal de Chomsky.

Para ejemplificar el funcionamiento del algoritmo tomamos $x = abba$ y G como la gramática

$$\begin{array}{lll} S & \longrightarrow & A\,B \mid B\,A \mid S\,S \mid A\,C \mid B\,D, \\ A & \longrightarrow & a, \qquad B \longrightarrow b, \\ C & \longrightarrow & S\,B, \qquad D \longrightarrow S\,A. \end{array}$$

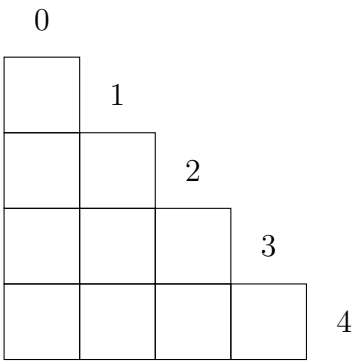
Considérense los siguientes separadores de x:

$$\underset{0}{|}x_1\underset{1}{|}x_2\underset{2}{|}\dots\underset{n-1}{|}\quad x_n\quad\underset{n}{|}$$

Así es posible denotar subcadenas como $x_{i,j} = x_{i+1} \dots x_j$, con $0 \le i < j \le n$. Para nuestra $x = abba$ tenemos $\underset{0}{|}a\underset{1}{|}b\underset{2}{|}b\underset{3}{|}a\underset{4}{|}$, algunas de sus subcadenas son:

- $x_{0,4} = abba$. La cadena entera se denota $x_{0,n}$, en este caso $x_{0,4}$.
- $x_{1,3} = bb$
- $x_{1,4} = bba$

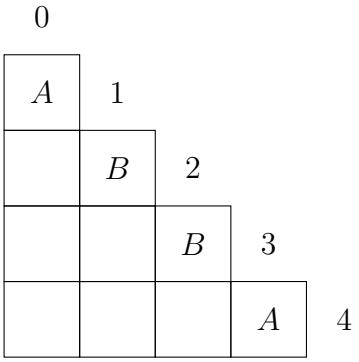
También se requiere de una tabla T triangular inferior de tamaño $\binom{n+1}{2}$, donde cada celda $T_{i,j}$ hace referencia a la subcadena $x_{i,j}$ y representa *el conjunto de variables que generan la cadena* $x_{i,j}$. Para nuestro ejemplo la tabla T tiene la forma:



*Esta nota se basa en el libro: D. C. Kozen. *Automata and Computability*, Springer-Verlag, Inc.

El algoritmo iniciará con subcadenas de x de tamaño uno, es decir, las subcadenas serán los símbolos de x, y buscará las variables de G que tengan a tales símbolos del lado derecho de sus producciones. Luego, para subcadenas de tamaño mayor a uno, el algoritmo las descompondrá en dos, de modo que se concatenen las dos variables que generan cada parte de la descomposición, y dicha concatenación se buscará en los lados derechos de las producciones de G. Lo anterior es porque la gramática G está en FNC, es decir, las producciones de G son de la forma $A \longrightarrow a$ y $A \longrightarrow XY$, con $A, X, Y \in V$ y $a \in T$. Por lo que no tiene sentido descomponer las subcadenas de x en tres, cuatro, etc., realizando el procedimiento descrito antes basta.

Comenzamos con las subcadenas de tamaño uno. Estas son las subcadenas de x de la forma $x_{i,i+1}$ para $0 \leq i \leq n-1$ y corresponden a las celdas de la tabla en la diagonal principal. Para cada subcadena $c = x_{i,i+1}$, si hay una producción $X \longrightarrow c \in G$, escribimos la variable X en la celda $T_{i,i+1}$. En nuestro ejemplo llegamos a lo siguiente:



En la celda $T_{1,2}$ escribimos a la variable B porque $x_{1,2} = b$ y $B \longrightarrow b$ es una producción de G. En general $T_{i,i+1}$ puede contener muchas variables, porque puede haber muchas producciones diferentes con $x_{i,i+1}$ en el lado derecho. Todas estas variables las escribimos en la celda $T_{i,i+1}$.

Procedemos ahora con las subcadenas de longitud dos, $x_{i,i+2}$. Estas corresponden a la diagonal en T abajo de la diagonal principal que acabamos de llenar. Para cada subcadena $x_{i,i+2}$ descomponemos la cadena en dos subcadenas $x_{i,i+1}$ y $x_{i+1,i+2}$ de longitud uno cada una y consideramos las celdas $T_{i,i+1}$ y $T_{i+1,i+2}$ correspondientes a esas subcadenas. Seleccionamos una variable de cada una de estas celdas, digamos X de $T_{i,i+1}$ y Y de $T_{i+1,i+2}$, y buscamos si hay producciones $Z \longrightarrow XY$ en G. Para cada una de tales producciones que encontremos, etiquetamos $T_{i,i+2}$ con Z. Hacemos esto por cada una de las posibles elecciones de $X \in T_{i,i+1}$ y $Y \in T_{i+1,i+2}$.

En nuestro ejemplo, para $x_{0,2} = ab$, encontramos únicamente a A en $T_{0,1}$ y B en $T_{1,2}$, por lo que buscamos producciones con AB en el lado derecho. Encontramos $S \longrightarrow AB$ así que etiquetamos $T_{0,1}$ con S.

Para $T_{1,3}$, encontramos B inmediatamente arriba de esta celda y B a la derecha, por lo que buscamos producciones con BB en el lado derecho. No hay ninguna, de modo que $T_{1,3} = \varnothing$. Continuamos de este modo hasta llenar todas las celdas $T_{i,i+2}$.

0				
A	1			
S	B	2		
	$\varnothing$	B	3	
		S	A	4

Procedemos con subcadenas de longitud tres. Para cada una de ellas hay dos formas de descomponerla en subcadenas no vacías. Por ejemplo,

$$x_{0,3} = x_{0,1}x_{1,3} = x_{0,2}x_{2,3}.$$

Tenemos que analizar ambas posibilidades. Para la primera, tenemos $A \in T_{0,1}$ y $\varnothing$ en $T_{1,3}$, lo que indica que esta posibilidad se cierra. Analizamos ahora $T_{0,2}$ y $T_{2,3}$. Tenemos $S \in T_{0,2}$ y $B \in T_{2,3}$, así que buscamos producciones con SB en el lado derecho, tenemos $C \longrightarrow SB$, así etiquetamos a $T_{1,3}$ con C.

Para $x_{1,4} = x_{1,2}x_{2,4} = x_{1,3}x_{3,4}$, tenemos $B \in T_{1,2}$ y $S \in T_{2,4}$, así que buscamos producciones con BS en el lado derecho, pero no hay ninguna. También vemos que $\varnothing \in T_{1,3}$ y $A \in T_{3,4}$, lo que indica que esta posibilidad se cierra. No encontramos una variable que genere a $x_{1,4}$, por lo tanto $T_{1,4}$ se etiqueta como $\varnothing$. Nuestra tabla queda como se muestra enseguida:

0				
A	1			
S	B	2		
C	$\varnothing$	B	3	
	$\varnothing$	S	A	4

Continuamos con la subcadena de tamaño cuatro, que es la cadena original. Analizamos las tres formas de descomposición que tiene, $x_{0,4} = x_{0,1}x_{1,4} = x_{0,2}x_{2,4} = x_{0,3}x_{3,4}$, obteniendo como resultado final la tabla

0				
A	1			
S	B	2		
C	$\varnothing$	B	3	
S	$\varnothing$	S	A	4

Al término del algoritmo, se analiza el conjunto en la celda $T_{0,n}$. Si el símbolo inicial S está en ese conjunto, entonces $x \in L(G)$. En nuestro ejemplo, como $T_{0,4}$ contiene a S, sabemos que $S \Rightarrow^* x_{0,4} = x$, concluimos que x es generada por G.

En este ejemplo en particular hay a los más una variable en cada celda. Ya que las variables de la gramática de la que partimos generan lenguajes disjuntos. En general, puede haber más de una variable en cada celda.

Abajo se muestra una descripción formal del algoritmo. Podemos corroborar por la estructura de ciclo anidada que la complejidad del algoritmo es $O(pn^3)$, donde $n = |x|$ y p es el número de producciones de G.

Algorithm 1.1: CKY, ejemplo de programación dinámica.

```
1  for i := 0, ..., n − 1 do                          /* primero cadenas de longitud 1 */
2      T_{i,i+1} := ∅                                                  /* inicializa a ∅ */
3      for A → a producción de G do
4          if a = x_{i,i+1} then
5              T_{i,i+1} := T_{i,i+1} ∪ {A}
6  for m := 2, ..., n do                                    /* por cada longitud m ≥ 2 */
7      for i := 0, ..., n − m do                  /* por cada subcadena de longitud m */
8          T_{i,i+m} := ∅                                              /* inicializa a ∅ */
9          for j := i + 1, ..., i + m − 1 do      /* para toda descomposición de x_{i,i+m} */
10             for A → BC producción de G do
11                 if B ∈ T_{i,j} ∧ C ∈ T_{j,i+m} then
12                     T_{i,i+m} := T_{i,i+m} ∪ {A}
```

Ejercicios 1 Dada la siguiente gramática libre de contexto G en FNC, determine mediante el algoritmo CKY si la cadena $w = aaabbb$ pertenece a $L(G)$.

$$G: \quad \begin{array}{lcll} S & \longrightarrow & AB \,|\, XB, & \\ X & \longrightarrow & AS, & \\ A & \longrightarrow & a, \qquad B \longrightarrow b. & \end{array}$$