

INF155: Informática Teórica

Clase 15: De PDA a CFG

Aldo Berrios Valenzuela

Martes 26 de Abril de 2016

1. PDA \rightarrow CFG

Teorema 1.1 (PDA \rightarrow CFG). Sea $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$ un PDA. Entonces, hay una gramática de contexto libre $G = (N, \Sigma, P, S)$ tal que $\mathcal{L}(G) = \mathcal{N}(M)$.

Demostración (informal). La idea general de la demostración consiste en emular una gramática de contexto libre a través de un PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$ que acepte por stack vacío. Para ello, definimos la gramática $G = (N, \Sigma, P, S)$ tal que:

1. S corresponde al símbolo de partida.
2. Los no-terminales son $[p, A, q]$, con $p, q \in Q$, y $A \in \Gamma$.

Las producciones de G son las siguientes:

- a) Para todos los estados $p \in Q$, G tiene producciones

$$S \rightarrow [q_0, Z_0, p]$$

Lo que se busca es eliminar Z_0 del stack. Para hacerlo, es necesario pasar por una montonera de estados: comenzamos en q_0 y terminamos en p . En otras palabras, $[q_0, Z_0, p]$ es todo string $\sigma \in \Sigma^*$ tal que M saque a Z_0 del stack, es decir

$$(q_0\sigma, Z_0) \vdash_M^* (\sigma p, \epsilon)$$

- b) Si $(p, B_1 B_2 \dots B_n) \in \delta(q, A, x)$, con $x \in \Sigma \cup \{\epsilon\}$. Entonces, para toda una lista de estados q_1, q_2, \dots, q_n , G tiene la producción:

$$[q, A, q_n] \rightarrow x[p, B_1, q_1][q_1, B_2, q_2] \dots [q_{n-1}, B_n, q_n] \quad (1.1)$$

La producción (1.1) dice que la única forma de sacar A del stack e ir desde el estado q a q_n es leer x . De inmediato, usar alguna entrada para sacar B_1 del stack mientras vamos del estado p al estado q_1 . En seguida, remover B_2 del stack a medida que vamos de q_1 a q_2 , y así sucesivamente hasta sacar B_n del stack mientras vamos del estado q_{n-1} a q_n .

Nótese que (1.1) representa sólo una movida posible del PDA... considere las demás.

- c) Si $(p, \epsilon) \in \delta(q, A, x)$, con $x \in \Sigma \cup \{\epsilon\}$, entonces:

$$[q, A, x] \rightarrow x$$

Importante destacar que si $x = \epsilon$, esto no es CFG. En breve corregiremos esto.

□

1.1. Elimino Producciones ϵ

Pueden permitirse producciones $A \rightarrow \epsilon$ en CFG (y en gramáticas regulares).

Demostración. La idea de la demostración consiste en identificar no-terminales $A \in N$ tales que $A \Rightarrow^* \epsilon$; “saltarse” esos pasos agregando versiones adicionales de los lados derechos donde aparece A sin aparecer esa A ([considere que el lenguaje original es de una gramática de tipo 0](#)).

Si $S \Rightarrow^* \epsilon$, muestra gramática modificada genera $L \setminus \{\epsilon\}$ (no digamos que es un cambio tan substancial...)

- Identificar no-terminales $A \Rightarrow^* \epsilon$ (“nullable symbols”)

“Inducción”: para construir el conjunto \mathcal{A} (es el conjunto de nullable symbols).

Base: Si $A \rightarrow \epsilon$ es una producción, $A \in \mathcal{A}$

Inducción: Si $A \rightarrow B_1 B_2 \dots B_n$ es una producción, y todos los $B_i \in \mathcal{A}$, entonces $A \in \mathcal{A}$.

- Cambiar la gramática:
 - Eliminar producciones $A \rightarrow \epsilon$
 - Producciones $A \rightarrow \alpha B \beta$, con $B \in \mathcal{A}$ agregar $A \rightarrow \alpha \beta$ (siempre que $\alpha \beta \neq \epsilon$)
 - Opcional: si realmente quiere mantener el mismo lenguaje, agregue $S \rightarrow \epsilon$. Más formal, si $S \Rightarrow^* \epsilon$ ($S \in \mathcal{A}$), agregar $S \rightarrow \epsilon$.

Ejemplo:

$$\begin{aligned} S &\rightarrow AB \\ A &\rightarrow aAA \mid \epsilon \\ B &\rightarrow bBB \mid \epsilon \end{aligned}$$

Primer paso: identificar no terminales que dan ϵ :

$$\mathcal{A}^{(0)} = \{A, B\}$$

Dado que S nos lleva a $A, B \in \mathcal{A}^{(0)}$, también debemos agregarlo al conjunto $\mathcal{A}^{(0)}$:

$$\mathcal{A}^{(1)} = \{A, B, S\}$$

Nuevamente, debemos revisar algunos de los conjuntos presentes en $\mathcal{A}^{(1)}$ nos llevan a producciones de la forma $X \rightarrow \epsilon$. Revisando nuevamente vemos que no hay más, por lo tanto:

$$\mathcal{A} = \{S, A, B\}$$

Cambiar la gramática:

$$\begin{aligned} S &\rightarrow AB \\ A &\rightarrow aAA \\ ~~A &\rightarrow \epsilon~~ \\ B &\rightarrow bBB \\ ~~B &\rightarrow \epsilon~~ \\ S &\rightarrow A & B &\Rightarrow^* \epsilon \\ S &\rightarrow B & A &\Rightarrow^* \epsilon \\ A &\rightarrow aA & A &\Rightarrow^* \epsilon, \text{ se fue la primera } A \text{ o se fue la segunda } A \text{ en } A \rightarrow aAA \\ A &\rightarrow a & A &\Rightarrow^* \epsilon, \text{ se nos va la } A \text{ dos veces en } A \rightarrow aAA \\ B &\rightarrow bB & B &\Rightarrow^* \epsilon, \text{ se fue la primera } B \text{ o se fue la segunda } B \text{ en } B \rightarrow bBB \\ B &\rightarrow b & B &\Rightarrow^* \epsilon, \text{ se nos va la } B \text{ dos veces en } B \rightarrow bBB \\ S &\rightarrow \epsilon & \text{Agregue esto sólo si quiere conservar el lenguaje original} \end{aligned}$$

Pasos siguientes:

- Eliminar símbolos inútiles (no aparecen nunca en ninguna producción).

En $G = (N, \Sigma, P, S)$ un símbolo $X \in N \cup \Sigma$ se dice útil si hay una derivación $S \Rightarrow^* \alpha X \beta \Rightarrow^* \sigma \in \Sigma^*$

- Propiedades de Clausura.

□