

# INF155: Informática Teórica

## Clase 11: Autómatas de Stack

Aldo Berrios Valenzuela

Martes 12 de Abril de 2016

### 1. Definiciones

---

**Definición 1.1.** Una descripción instantánea del PDA  $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$  es:

$$(\alpha q \beta, \gamma) \tag{1.1}$$

con  $\alpha, \beta \in \Sigma^*$  (entrada leída y por leer)  $q \in Q$  (estado actual),  $\gamma \in \Gamma^*$  (stack actual).

**Definición 1.2.**  $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$  un PDA. La relación de transición de  $M$ ,  $\vdash_M$  entre ID's de  $M$  se define por:

- Si  $(p, \sigma) \in \delta(q, \epsilon, A)$  entonces:

$$(\alpha q \beta, \gamma A) \vdash_M (\alpha p \beta, \gamma \sigma) \tag{1.2}$$

- Si  $(p, \sigma) \in \delta(q, a, A)$ :

$$(\alpha q a \beta, \gamma A) \vdash_M (\alpha p \beta, \gamma \sigma) \tag{1.3}$$

Se suele escribir  $\vdash$  si  $M$  se subentiende.

**Definición 1.3.** Sea  $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$  un PDA. El lenguaje aceptado por  $M$  por estado final es:

$$\mathcal{L}(M) = \{\sigma : (q_0 \sigma, Z_0) \vdash_M^* (\sigma q_f, \gamma) \wedge \gamma \in N \wedge q_f \in F\} \tag{1.4}$$

**Definición 1.4.** Sea  $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$  un PDA. El lenguaje aceptado por stack vacío por  $M$  es:

$$\mathcal{N}(M) = \{\sigma : (q_0 \sigma, Z_0) \vdash_M^* (\sigma q, \epsilon)\} \tag{1.5}$$

En general,  $\mathcal{L}(M) \neq \mathcal{N}(M)$  (los lenguajes aceptados por estado final y por stack vacío son diferentes). Al diseñar PDA para aceptar por stack vacío, por convención se da  $F = \emptyset$ .

**Definición 1.5.** El PDA  $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$  se dice determinista (DPDA) si:

- I. Para todo  $q \in Q$ ,  $x \in (\Sigma \cup \{\epsilon\})$ ,  $A \in \Gamma$  es  $|\delta(q, x, A)| \leq 1$ , es decir, el PDA tiene a lo más 1 movida posible.
- II. Si para todo  $a \in \Sigma$ ,  $\delta(q, a, A) \neq \emptyset$ , entonces  $\delta(q, \epsilon, A) = \emptyset$ . Con esto, forzamos al PDA para que tenga sólo una movida.

**Teorema 1.1.** Hay lenguajes aceptados por PDA que **no** son aceptados por DPDA.

*Demostración.* Consideremos el lenguaje de (1.6).

$$L = \{\sigma \sigma^R : \sigma \in \{a, b\}^*\} \tag{1.6}$$

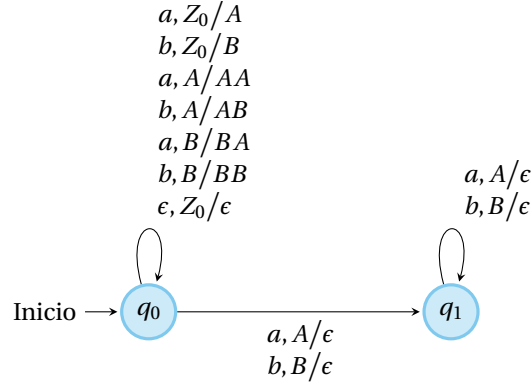


Figura 1: En el loop del estado  $q_0$  llenamos el stack del PDA. Luego, al pasar de  $q_0 \rightarrow q_1$  incluyendo el loop, vaciamos el stack.

La Figura 1 representa un PDA que reconoce  $L$  por stack vacío. Este PDA no es determinista:

$$\delta(q_0, a, A) = \{(q_0, AA), (q_1, \epsilon)\}$$

$$\delta(q_0, b, B) = \{(q_0, BB), (q_1, \epsilon)\}$$

Informalmente,  $\mathcal{N}(M) = L$ .

Pero ningún DPDA puede aceptar  $L$  por stack vacío. Como  $\epsilon \in L$ , después de la primera movida, un DPDA queda con stack vacío, y no consume el resto si  $\sigma \in L$  con  $\sigma \neq \epsilon$ .  $\square$

**Observación:** La demostración formal del teorema 1.1 se puede hacer por inducción. El caso base es la palabra  $\sigma$ .

**Teorema 1.2.** Si  $L = \mathcal{L}(M)$ , para  $M$  un PDA, podemos construir un PDA  $M'$  tal que  $L = \mathcal{N}(M')$ .

*Demostración.* La idea consiste en consumir el stack al llegar a un estado final. Evitar aceptación “accidental” por  $Z'_0$  bajo el stack de  $M$ .

Sea  $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ . Definimos  $M' = (Q \cup \{q'_0, q'_f\}, \Sigma, \Gamma, \delta', q'_0, Z'_0, \emptyset)$ , donde:

$$\delta'(q'_0, \epsilon, Z'_0) = \{(q_0, Z'_0 Z_0)\} \quad (1.7)$$

Todos las movidas de  $M$ :

$$\delta'(q, x, A) = \delta(q, x, A) \quad (1.8)$$

siempre que  $q \neq F$

$$\delta'(q_f, \epsilon, A) = \delta(q_f, \epsilon, A) \cup \{(q'_f, \epsilon)\} \quad (1.9)$$

para  $q_f \in F$ . Además, para todo lo anterior considere  $x \in \Sigma \cup \{\epsilon\}$ ,  $q \in Q \setminus F$ ,  $q_f \in F$ ,  $A \in \Gamma$ ,  $a \in \Sigma$ .

$$\delta'(q'_f, \epsilon, A) = \{(q'_f, \epsilon)\} \text{ para } A \in \Gamma \cup \{Z'_0\} \quad \square$$

**Teorema 1.3.** Si  $L = \mathcal{N}(M)$ , podemos constuir  $M'$  con  $L = \mathcal{L}(M')$

*Demostración.* Sea  $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ . Definimos  $M' = (Q \cup \{q'_0, q'_f\}, \Sigma, \Gamma \cup \{Z'_0\}, \delta', q'_0, Z'_0, \{q'_f\})$  con:

$$\begin{aligned} \delta'(q'_0, Z'_0) &= \{(q_0, Z'_0 Z_0)\} \\ \delta'(q, x, A) &= \delta(q, x, A) && \text{para } q \in Q, x \in \Sigma \cup \{\epsilon\}, A \in \Gamma \\ \delta'(q, \epsilon, Z'_0) &= \{(q'_f, \epsilon)\} && \text{para } q \in Q \end{aligned}$$

$\square$