# Capítulo 8

# Automatas de pila

## 8.1. Introducción

Los autómatas de pila son autómatas que cuentan con las mismas características que un $AEFND$ pero ademas disponen de una pila de memoria en la que podrán leer o escribir símbolos, de forma de poder saber en cada momento si el autómata ya realizo alguna transición en el pasado. Durante cada transición el autómata ejecuta la siguiente secuencia:

1. Leer un símbolo de entrada.
2. Extraer un símbolo de la pila.
3. Insertar un símbolo en la pila.
4. Pasar a un nuevo estado.

A este proceso lo representaremos con la notación $(\sigma, l, \alpha; \beta, \tau)$ donde:

- $\sigma$ es el estado actual.
- $l$ es el símbolo del alfabeto que se lee en la entrada.
- $\alpha$ es el símbolo que se extrae de la pila.
- $\beta$ es el símbolo que se inserta en la pila.
- $\tau$ es el nuevo estado al que pasa el autómata.

Representaremos este proceso mediante el siguiente diagrama de transiciones:

$$\sigma \xrightarrow{l,\alpha;\beta} \tau$$

Puesto que permitimos que $l, \alpha, \beta$ sean la cadena vacía, podemos en cualquier transición no leer un caracter, no extraer nada de la pila o no escribir nada en la pila si así lo necesitáramos. Llamaremos a las transiciones de la forma $(\sigma, \lambda, \lambda; \lambda, \tau)$ transiciones espontaneas.

**Observación** Nótese que las transiciones representada por $(\sigma, l, \lambda; \lambda, \tau)$ son las que realiza un $AEFND$. Por lo tanto los $AEFND$ son un caso particular de $AP$ y en consecuencia los lenguajes regulares son un subconjunto de los lenguajes aceptados por $AP$, es decir: $\mathcal{L}_3 \subseteq \mathcal{AC}\,(AP)$.

## 8.2. Definición formal

Un autómata de pila $(AP)$ es una sextupla $A = (S, \Sigma, \Gamma, T, \sigma, Ac)$ donde:

- $S$ es un conjunto finito de *estados*.
- $\Sigma$ es un conjunto finito de *símbolos de entrada*.
- $\Gamma$ es un conjunto finito de *símbolos de pila*.
- $T \subseteq S \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \times S$ es una *relación de transición*.
- $\sigma \in S$ es el *estado inicial*.
- $Ac \subseteq S$ es un conjunto de estados de aceptación.

## 8.3. Configuración

Una configuración de un autómata de pila $A = (S, \Sigma, \Gamma, T, \sigma, Ac)$ es un elemento de $\mathcal{C}_A = S \times \Sigma^* \times \Gamma^*$. Dicha configuración $(\sigma, p, \gamma)$ indica que el autómata esta en el estado $\sigma$, le falta leer la cadena $p$ de la entrada y el contenido completo de la pila es $\gamma$.

## 8.4. Relación entre configuraciones

Para una autómata de pila $A = (S, \Sigma, \Gamma, T, \sigma, Ac)$ definimos la relación «lleva en un paso» (que notaremos $\Rightarrow_A$) entre configuraciones, de la siguiente forma: $(\sigma, lp, \alpha\gamma) \Rightarrow_A (\tau, p, \beta\gamma)$ si y solo si $(\sigma, l, \alpha; \beta, \tau) \in T$. La relación «lleva en uno o mas pasos» $(\Rightarrow_A^*)$ define recursivamente a partir de la relación $\Rightarrow_A$ de forma análoga a lo hecho para la función de transición $f$ de los $AEF$.

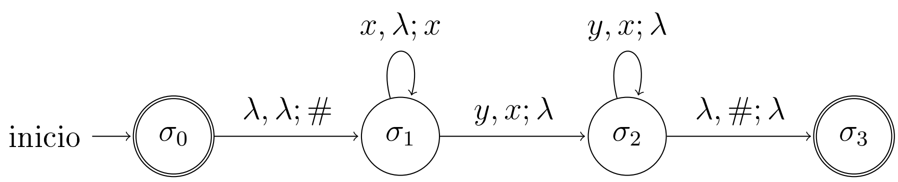
## 8.5. Lenguaje aceptado

Sea $A = (S, \Sigma, \Gamma, T, \sigma, Ac)$ un autómata de pila, el lenguaje aceptado por $A$ es el conjunto: $\mathcal{AC}(A) = \{p \in \Sigma^* / (\sigma, p, \lambda) \Rightarrow^* (\tau, \lambda, \gamma) : \gamma \in \Gamma^* \wedge \tau \in Ac\}$. Es decir, una cadena $p$ sera aceptada por un $AP$ si, arrancando desde su estado inicial y con la pila vaciá, es posible que el autómata llegue a un estado de aceptación después de leer toda la cadena.

**Observación** No necesariamente se llegara a un estado de aceptación luego de leer el ultimo caracter de una palabra pues a continuación el autómata podría realizar transiciones de la forma $(\sigma, \lambda, \alpha; \beta, \tau)$ y llegar luego al estado de aceptación.

## 8.6. Aceptacion de lenguajes no regulares

Ya hemos visto que $\mathcal{L}_3 \subseteq \mathcal{AC}(AP)$. Sin embargo esta inclusión es estricta, pues como veremos, el lenguaje $\{x^n y^n / n \in \mathbb{N}_0\}$ es aceptado por el siguiente autómata de pila:



## 8.7. Teorema del vaciado de pila

**Enunciado** Para cada $A \in AP$ existe $A' \in AP$ tal que $A'$ vaciá su pila y ademas $\mathcal{AC}(A') = \mathcal{AC}(A)$.

**Demostración** Sea $A = (S, \Sigma, \Gamma, T, \sigma, Ac)$ un $AP$, fabricaremos $A'$ de la siguiente manera:

- El estado inicial de $A$ deja de serlo, pues introducimos un nuevo estado inicial y una transición del nuevo al anterior que lo único que hace es insertar en la pila un marcador # (suponiendo que $\# \notin \Gamma$).
- Los estados de aceptación de $A$ dejan de serlo e introduciremos un nuevo estado $P$ junto con transiciones espontaneas que pasan de cada uno de los antiguos estados de aceptación al nuevo estado $P$.
- Vaciamos la pila sin salir del estado $P$ introduciendo transiciones de la forma $(P, \lambda, x; \lambda, P)$ para cada $x \in \Gamma$.
- Agregamos un nuevo y único estado de aceptación $Q$ junto a la transición $(P, \lambda, \#; \lambda, Q)$.

Formalmente definimos $A' = (S', \Sigma, \Gamma', T', \sigma', Ac')$ donde:

- $S' = S \cup \{R, P, Q\}$ donde $R, P, Q \notin S$.
- $\Gamma' = \Gamma \cup \{\#\}$ donde $\# \notin \Gamma$.
- $\sigma'_0 = R$.
- $Ac' = \{Q\}$.
- $$\begin{aligned} T' = T &\cup \{(R, \lambda, \lambda; \#, \sigma)\} && (1) \\ &\cup \{(\tau, \lambda, \lambda; \lambda, P) / \tau \in Ac\} && (2) \\ &\cup \{(P, \lambda, \alpha; \lambda, P) / \alpha \in \Gamma\} && (3) \\ &\cup \{(P, \lambda, \#; \lambda, Q)\}. && (4) \end{aligned}$$

Veamos ahora que $\mathcal{AC}(A) = \mathcal{AC}(A')$:

- $\subseteq$: Sea $\alpha \in \mathcal{AC}(A)$. Sabemos que partiendo del estado inicial $\sigma$ con la pila vaciá, $\alpha$ nos lleva en uno o mas pasos a un estado de aceptación. En términos de configuraciones: $(\sigma, \alpha, \lambda) \Rightarrow^* (\tau, \lambda, \gamma)$ donde $\tau \in Ac, \gamma \in \Gamma^*$. Luego en $A'$ tenemos la derivación:

  $$(R, \alpha, \lambda) \Rightarrow^{(1)} (\sigma, \alpha, \#) \Rightarrow^* (\tau, \lambda, \gamma\#) \Rightarrow^{(2)} (P, \lambda, \gamma\#) \Rightarrow^{(3)*} (P, \lambda, \#) \Rightarrow^{(4)} (Q, \lambda, \lambda)$$

  Como $Q \in Ac'$ concluimos que $\alpha \in \mathcal{AC}(A')$ y la pila queda vacía.

- $\supseteq$: Análogo.

## 8.8. Igualdad entre $\mathcal{L}_2$ y $\mathcal{AC}(AP)$

**Enunciado**

1. Sea $G$ una gramática independiente de contexto entonces existe un autómata de pila $A$ tal que $L(G) = \mathcal{AC}(A)$.

2. Sea $A$ un autómata de pila entonces existe una gramática independiente del contexto $G$ tal que $L(G) = \mathcal{AC}(A)$.

Es decir $\mathcal{L}_2 = \mathcal{AC}(AP)$.

**Demostración**

1. Consultar «J. Glenn Brookshear. *Teoría de la Computación. Lenguajes formales, autómatas y complejidad*», pag 85.

2. Consultar «J. Glenn Brookshear. *Teoría de la Computación. Lenguajes formales, autómatas y complejidad.*», pag 90.

**Conclusión** De todo lo que hemos visto hasta ahora sabemos que:

$$\mathcal{L}_3 = \mathcal{AC}(AEF) = \mathcal{AC}(AEFND) = L_{ER} \subset \mathcal{L}_2 = \mathcal{AC}(AP)$$

## 8.9. Lema de bombeo para autómatas de pila

**Enunciado** Sea $L \in \mathcal{L}_2$, luego si $L$ es infinito existe $p \in L$ de la forma $p = xuyvz$ donde $uv \neq \lambda$ tal que $xu^n yv^n z \in L\ \forall n \in \mathbb{N}$.

**Demostración** Sabemos que existe una gramática independiente de contexto $G = (N, T, P, \sigma)$ tal que $L(G) = L$ y sea $m$ el máximo numero de símbolos de $N \cup T$ que aparecen en el lado derecho de las reglas de producción de $G$. Observemos que como $m$ es la mayor cantidad de símbolos por los que se puede reemplazar un no terminal, al aplicar $i$ reglas de producción cualesquiera, la longitud de la cadena resultante es a lo sumo $m^i$. Llamemos $k$ a la cantidad de símbolos no terminales de $G$ ($k = |N|$). Dada una palabra $p \in L$ tal que $|p| > m^k$ (que existe pues $L$ es infinito), llamemos $i$ a la cantidad de reglas de producción aplicadas para producir $p$. Por lo tanto $m^k \leq |p| \leq m^i \Rightarrow k < i$ es decir, para formar $p$ debieron aplicarse mas de $k$ reglas de producción (debieron expandirse mas de $k$ no terminales). Como no hay $k$ no terminales, existe un no terminal $X$ que debió expandirse 2 veces que podremos «bombear» cuantas veces sea necesario.

## 8.10. Existencia de lenguajes sensibles al contexto

**Enunciado** Existen lenguajes sensibles al contexto.

**Demostración** Probaremos que $L = \{a^n b^n c^n / n \in \mathbb{N}\} \notin \mathcal{L}_2$. Como $L$ es infinito el lema de bombeo nos permite asegurar que existe una cadena $xuyvz \in L$ tal que $xu^n yv^n z \in L\ \forall n \in \mathbb{N}$ con $uv \neq \lambda$. Supongamos sin perder generalidad que $u \neq \lambda$ luego hay dos posibilidades:

- $u$ esta formado por un solo símbolo y al bombearlo se obtiene una palabra que no mantiene la igualdad entre exponentes resultando no pertenecer al lenguaje.

- $u$ esta formado por mas de un caracter y al bombearlo se obtiene una palabra que altera el orden de los símbolos y por lo tanto no pertenece al lenguaje.

Llegamos al absurdo por cualquiera de las dos posibilidades, por lo que $L \notin \mathcal{L}_2$.