# Capítulo 5

# Autómatas con Pila

En el presente capítulo presentamos el modelo de autómata requerido para aceptar los lenguajes independientes del contexto: el autómata con pila no-determinista. Existe también la versión determinista pero, a diferencia de lo que sucede con los modelos AFD y AFN, los modelos de autómata con pila determinista y no-determinista no resultan ser computacionalmente equivalentes.

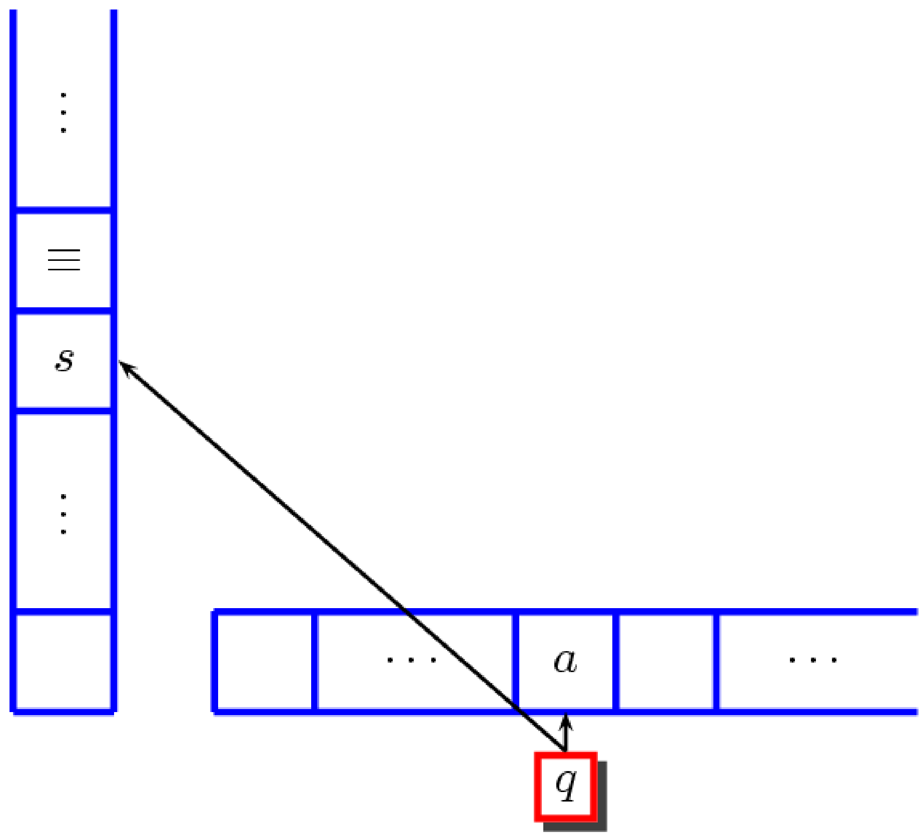
## 5.1. Autómatas con Pila Deterministas (AFPD)

Un **Autómata Finito con Pila Determinista** (AFPD) es una 7-upla, $M = (Q, q_0, F, \Sigma, \Gamma, s_0, \Delta)$, con los siguientes componentes:

1. $Q$ es el conjunto (finito) de estados.
2. $q_0 \in Q$ es el estado inicial.
3. $F$ es el conjunto de estados finales o de aceptación, $\varnothing \neq F \subseteq Q$.
4. $\Sigma$ es el alfabeto de cinta.
5. $\Gamma$ es el alfabeto de pila.
6. $s_0 \in \Gamma$ es el símbolo inicial de pila.
7. $\Delta$ es la función de transición del autómata:

$$\Delta : Q \times (\Sigma \cup \lambda) \times \Gamma \to (Q \times \Gamma^*).$$

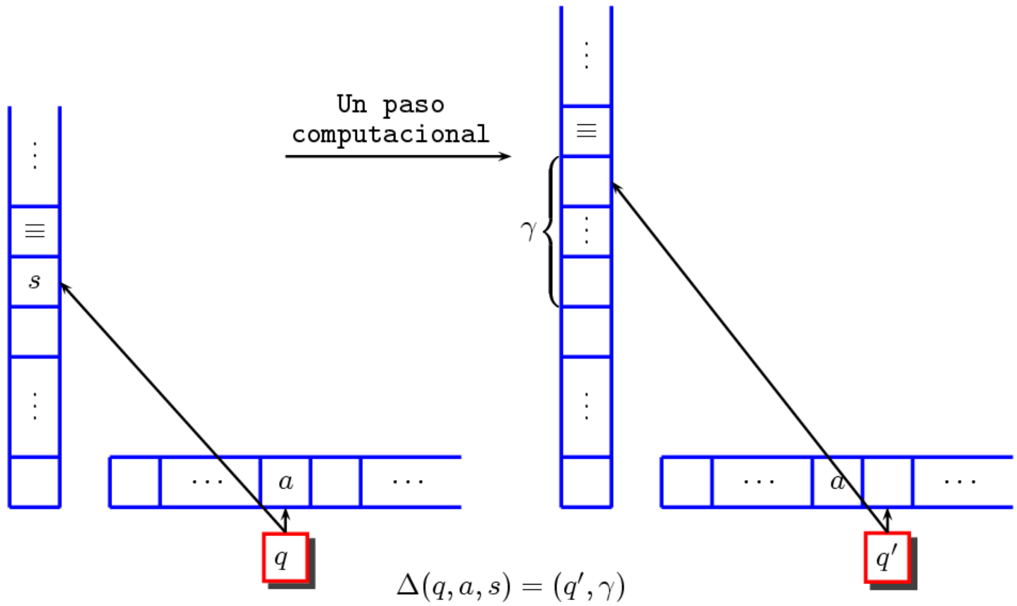
Como en los modelos ya considerados (AFD, AFN y AFN-$\lambda$), un AFPD procesa cadenas sobre una cinta de entrada semi-infinita, pero hay una cinta adicional, llamada pila, que es utilizada por el autómata como lugar de almacenamiento. En un momento determinado, la unidad de control del autómata escanea un símbolo $a$ sobre la cinta de entrada y el símbolo $s$ en el tope o cima de la pila, como lo muestra la siguiente gráfica:




La transición

$$\Delta(q, a, s) = (q', \gamma)$$

representa un **paso computacional**: la unidad de control pasa al estado $q'$ y se mueve a la derecha; además, borra el símbolo $s$ que está en el tope de la pila, escribe la cadena $\gamma$ (cadena que pertenece a $\Gamma^*$) y pasa a escanear el nuevo tope de la pila. La gráfica que aparece en la parte superior de la página siguiente ilustra un paso computacional. Recalcamos que en un paso computacional, el autómata sólo tiene acceso al símbolo que está en el tope de la pila; además, el contenido de la pila siempre se lee desde arriba (el tope) hacia abajo. Por estas dos razones la pila se dibuja verticalmente.




$$\Delta(q, a, s) = (q', \gamma)$$

**Casos especiales de transiciones:**

1. $\Delta(q, a, s) = (q', s)$. En este caso, el contenido de la pila no se altera.

2. $\Delta(q, a, s) = (q', \lambda)$. El símbolo $s$ en el tope de la pila se borra y el control finito pasa a escanear el nuevo tope de la pila, que es el símbolo colocado inmediatamente debajo de $s$.

3. $\Delta(q, \lambda, s) = (q', \gamma)$. Ésta es una transición $\lambda$: el símbolo sobre la cinta de entrada no se procesa y la unidad de control no se mueve a la derecha, pero el tope $s$ de la pila es reemplazado por la cadena $\gamma$. Para garantizar el determinismo, $\Delta(q, a, s)$ y $\Delta(q, \lambda, s)$, con $a \in \Sigma$, no pueden estar simultáneamente definidos (de lo contrario el autómata tendría una opción no-determinista). Las transiciones $\lambda$ en un AFPD permiten que el autómata cambie el contenido de la pila, sin procesar (o consumir) símbolos sobre la cinta de entrada.

**Configuración o descripción instantánea.** Es una tripla $(q, au, s\beta)$ que representa lo siguiente: el autómata está en el estado $q$, $au$ es la parte no procesada de la cadena de entrada, y la unidad de control está escaneando el símbolo $a$. La cadena $s\beta$ es el contenido *total* de la pila; siendo $s$ el símbolo colocado en el tope.

La notación $(q, au, s\beta)$ para configuraciones instantáneas es muy cómoda: para representar el paso computacional de la figura que aparece arriba escribimos simplemente

$$(q, au, s\beta) \vdash (p, u, \gamma\beta).$$

Aquí el autómata utilizó la transición $\Delta(q, a, s) = (p, \gamma)$.

La notación

$$(q, u, \beta) \overset{*}{\vdash} (p, v, \gamma)$$

significa que el autómata pasa de la configuración instantánea $(q, u, \beta)$ a la configuración instantánea $(p, v, \gamma)$ en cero, uno o más pasos computacionales.

**Configuración inicial.** Para una cadena de entrada $w \in \Sigma^*$, la configuración inicial es $(q_0, w, s_0)$. Al comenzar el procesamiento de toda cadena de entrada, el contenido de la pila es $s_0$, que sirve como marcador de fondo.

**Configuración de aceptación.** La configuración $(p, \lambda, \beta)$, siendo $p$ un estado final o de aceptación, se llama configuración de aceptación. Esto significa que, para ser aceptada, una cadena de entrada debe ser procesada completamente, con el control finito en un estado de aceptación. La cadena $\beta$ que queda en la pila puede ser cualquier cadena de símbolos en $\Gamma^*$.

**Lenguaje aceptado por un AFPD.** El lenguaje aceptado por un AFPD $M$ se define como

$$L(M) := \{w \in \Sigma^* : (q_0, w, s_0) \overset{*}{\vdash} (p, \lambda, \beta),\ p \in F\}$$

O sea, una cadena es aceptada si se puede ir desde la configuración inicial hasta una configuración de aceptación, en cero, uno o más pasos.

- ✎ En el modelo AFPD se permite que la transición $\Delta(q, a, s)$ no esté definida, para algunos valores $q \in Q$, $a \in \Sigma$, $s \in \Gamma$. Esto implica que el cómputo de algunas cadenas de entrada puede abortarse sin que se procesen completamente.
- ✎ No se debe confundir la tripla que aparece en la función de transición $\Delta(q, a, s)$ con la tripla $(q, u, \beta)$ que representa una configuración instantánea.
- ✎ La definición de la función de transición $\Delta$ requiere que haya por lo menos un símbolo en la pila. No hay cómputos con pila vacía.
- ✎ Para los autómatas con pila se pueden hacer diagramas de estados, similares a los ya conocidos, pero resultan de poca utilidad práctica ya que el procesamiento completo de una cadena de entrada depende del contenido de la pila, el cual puede cambiar en cada paso computacional.
- ✎ Los analizadores sintácticos en compiladores se comportan generalmente como autómatas con pila deterministas.

Un AFPD puede simular un AFD simplemente ignorando la pila; de esto se deduce que los lenguajes regulares son aceptados por autómatas AFPD. El siguiente teorema establece formalmente este resultado.

**5.1.1 Teorema.** *Todo lenguaje regular $L$ es aceptado por algún AFPD.*

*Demostración*: Sea $M = (Q, q_0, F, \Sigma, \delta)$ un AFD que acepta a $L$. El AFPD $M' = (Q, q_0, F, \Sigma, \Gamma, s_0, \Delta)$ definido haciendo $\Gamma = \{s_0\}$ y

$$\Delta(q, a, s_0) = (\delta(q, a), s_0), \text{ para todo } a \in \Sigma, \ q \in Q,$$

satisface claramente $L(M') = L(M) = L$. □

Sin usar la pila un AFPD no puede hacer nada más que un AFD, pero utilizando la pila como lugar de almacenamiento, un AFPD puede aceptar lenguajes no regulares, como se muestra en el siguiente ejemplo.

***Ejemplo*** Diseñar un AFPD que acepte el lenguaje $L = \{a^i b^i : i \geq 1\}$, sobre el alfabeto $\Sigma = \{a, b\}$. Recordemos que $L$ no es regular y no puede ser aceptado por ningún autómata normal (sin pila).

Solución: La idea es copiar las $a$es en la pila y borrar una $a$ por cada $b$ que sea leída sobre la cinta. Una cadena será aceptada si es procesada completamente y en la pila sólo queda el marcador de fondo $s_0$. Concretamente, $M = (Q, q_0, F, \Sigma, \Gamma, s_0, \Delta)$, donde

$$\begin{aligned}
\Sigma &= \{a, b\}, \\
\Gamma &= \{s_0, A, B\}, \\
Q &= \{q_0, q_1, q_2\}, \\
F &= \{q_2\},
\end{aligned}$$

y la función de transición está dada por:

$$\begin{aligned}
\Delta(q_0, a, s_0) &= (q_0, As_0), \\
\Delta(q_0, a, A) &= (q_0, AA), \\
\Delta(q_0, b, A) &= (q_1, \lambda), \\
\Delta(q_1, b, A) &= (q_1, \lambda), \\
\Delta(q_1, \lambda, s_0) &= (q_2, s_0).
\end{aligned}$$

Podemos ilustrar el procesamiento de varias cadenas de entrada. Sea, inicialmente, $u = aaabbb$.

$$\begin{aligned}
(q_0, aaabbb, s_0) &\vdash (q_0, aabbb, As_0) \vdash (q_0, abbb, AAs_0) \vdash (q_0, bbb, AAAs_0) \\
&\vdash (q_1, bb, AAs_0) \vdash (q_1, b, As_0) \vdash (q_1, \lambda, s_0) \vdash (q_2, \lambda, s_0).
\end{aligned}$$

La última es una configuración de aceptación; por lo tanto la cadena $u = aaabbb$ es aceptada.

Para la cadena de entrada $v = aabbb$, se obtiene el siguiente procesamiento:

$$
\begin{aligned}
(q_0, aabbb, s_0) &\vdash (q_0, abbb, As_0) \vdash (q_0, bbb, AAs_0) \vdash (q_1, bb, As_0) \\
&\vdash (q_1, b, s_0) \vdash (q_2, b, s_0). \quad [\text{cómputo abortado}]
\end{aligned}
$$

Obsérvese que el autómata ha ingresado al estado de aceptación $q_2$ pero la cadena de entrada no es aceptada debido a que no se ha procesado completamente; $(q_2, b, s_0)$ no es una configuración de aceptación.

Para la cadena de entrada $w = aaabb$, se tiene:

$$
\begin{aligned}
(q_0, aaabb, s_0) &\vdash (q_0, aabb, As_0) \vdash (q_0, abb, AAs_0) \vdash (q_0, bb, AAAs_0) \\
&\vdash (q_1, b, AAs_0) \vdash (q_1, \lambda, As_0).
\end{aligned}
$$

A pesar de que se ha procesado completamente la cadena de entrada $w$, la configuración $(q_0, \lambda, As_0)$ no es de aceptación. Por lo tanto, $w = aaabb$ no es aceptada.

***Ejemplo*** Diseñar un AFPD que acepte el lenguaje de todas las cadenas sobre el alfabeto $\Sigma = \{a, b\}$ (diferentes de $\lambda$) que tienen igual número de $a$es que de $b$es.

Solución: La idea es acumular las $a$es o $b$es consecutivas en la pila. Si en el tope de la pila hay una $A$ y el autómata lee una $b$, se borra la $A$; similarmente, si en el tope de la pila hay una $B$ y el autómata lee una $a$, se borra la $B$. La cadena de entrada será aceptada si es procesada completamente y en la pila sólo queda el marcador de fondo $s_0$. Concretamente, $M = (Q, q_0, F, \Sigma, \Gamma, s_0, \Delta)$, donde

$$
\begin{aligned}
\Sigma &= \{a, b\}, \\
\Gamma &= \{s_0, A, B\}, \\
Q &= \{q_0, q_1, q_2\}, \\
F &= \{q_2\},
\end{aligned}
$$

y la función de transición está dada por:

$$
\begin{aligned}
\Delta(q_0, a, s_0) &= (q_1, As_0),\\
\Delta(q_0, b, s_0) &= (q_1, Bs_0),\\
\Delta(q_1, a, s_0) &= (q_1, As_0),\\
\Delta(q_1, b, s_0) &= (q_1, Bs_0),\\
\Delta(q_1, a, A) &= (q_1, AA),\\
\Delta(q_1, b, B) &= (q_1, BB),\\
\Delta(q_1, a, B) &= (q_1, \lambda),\\
\Delta(q_1, b, A) &= (q_1, \lambda),\\
\Delta(q_1, \lambda, s_0) &= (q_2, s_0).
\end{aligned}
$$

A continuación procesamos algunas cadenas de entrada.

Cadena de entrada: $aabababb$ (aceptada).

$$
\begin{aligned}
(q_0, aabababb, s_0) &\vdash (q_1, abababb, As_0) \vdash (q_1, bababb, AAs_0)\\
&\vdash (q_1, ababb, As_0) \vdash (q_1, babb, AAs_0) \vdash (q_1, abb, As_0)\\
&\vdash (q_1, bb, AAs_0) \vdash (q_1, b, As_0) \vdash (q_1, \lambda, s_0) \vdash (q_2, \lambda, s_0).
\end{aligned}
$$

Cadena de entrada: $bbbaba$ (rechazada).

$$
\begin{aligned}
(q_0, bbbaba, s_0) &\vdash (q_1, bbaba, Bs_0) \vdash (q_1, baba, BBs_0) \vdash (q_1, aba, BBBs_0)\\
&\vdash (q_1, ba, BBs_0) \vdash (q_1, a, BBBs_0) \vdash (q_1, \lambda, BBs_0).
\end{aligned}
$$

En este último caso, la cadena de entrada $bbbaba$ es procesada completamente pero la configuración final no es de aceptación.

***Ejemplo*** Diseñar un AFPD que acepte el lenguaje

$$L = \left\{ wcw^R : w \in \{a, b\}^* \right\}.$$

sobre $\Sigma = \{a, b, c\}$. Nótese que las cadenas $w$ y $w^R$ sólo poseen $a$es y/o $b$es.

Solución: La idea es acumular los símbolos en la pila hasta que aparezca la $c$. Luego comparar los símbolos leídos con los almacenados en la pila, borrando en cada paso el tope de la pila. La cadena de entrada será aceptada si es procesada completamente y en la pila sólo queda el marcador de fondo $s_0$. Concretamente, $M = (Q, q_0, F, \Sigma, \Gamma, s_0, \Delta)$, donde

$$\begin{aligned}
\Sigma &= \{a, b, c\},\\
\Gamma &= \{s_0, A, B\},\\
Q &= \{q_0, q_1, q_2\},\\
F &= \{q_2\},
\end{aligned}$$

y la función de transición está dada por:

$$\begin{aligned}
\Delta(q_0, a, s_0) &= (q_0, As_0),\\
\Delta(q_0, b, s_0) &= (q_0, Bs_0),\\
\Delta(q_0, c, s_0) &= (q_2, s_0) \quad \text{(para aceptar la cadena } c\text{)},\\
\Delta(q_0, a, A) &= (q_0, AA),\\
\Delta(q_0, a, B) &= (q_0, AB),\\
\Delta(q_0, b, A) &= (q_0, BA),\\
\Delta(q_0, b, B) &= (q_0, BB),\\
\Delta(q_0, c, A) &= (q_1, A),\\
\Delta(q_0, c, B) &= (q_1, B),\\
\Delta(q_1, a, A) &= (q_1, \lambda),\\
\Delta(q_1, b, B) &= (q_1, \lambda),\\
\Delta(q_1, \lambda, s_0) &= (q_2, s_0).
\end{aligned}$$

## Ejercicios de la sección 5.1

1. Diseñar AFPD que acepten los siguientes lenguajes sobre $\Sigma = \{a, b\}$:

   (i) $L = \{a^i b^{2i} : i \geq 1\}$.

   (ii) $L = \{a^{2i} b^i : i \geq 1\}$.

   (iii) $L = \{a^i b^j : i \geq j \geq 1\}$.

2. Diseñar AFPD que acepten los siguientes lenguajes sobre $\Sigma = \{0, 1\}$:

   (i) $L = \{0^i 1^j 0^i : i, j \geq 1\}$.

   (ii) El lenguaje de las cadenas que tienen el doble de ceros que de unos.