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)

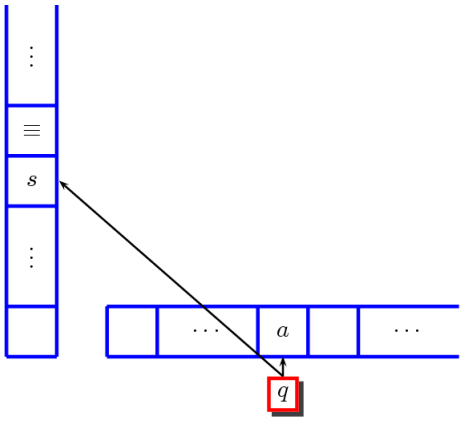
Un **Autómata Finito con Pila Determinista** (AFPD) es una 7-upla, $M = (Q, q_0, F, \Sigma, \Gamma, s_0, \Delta)$, con los siguientes componentes:

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. Σ es el alfabeto de cinta.
5. Γ es el alfabeto de pila.
6. $s_0 \in \Gamma$ es el símbolo inicial de pila.
7. Δ es la función de transición del autómata:

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

Como en los modelos ya considerados (AFD, AFN y AFN-λ), 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

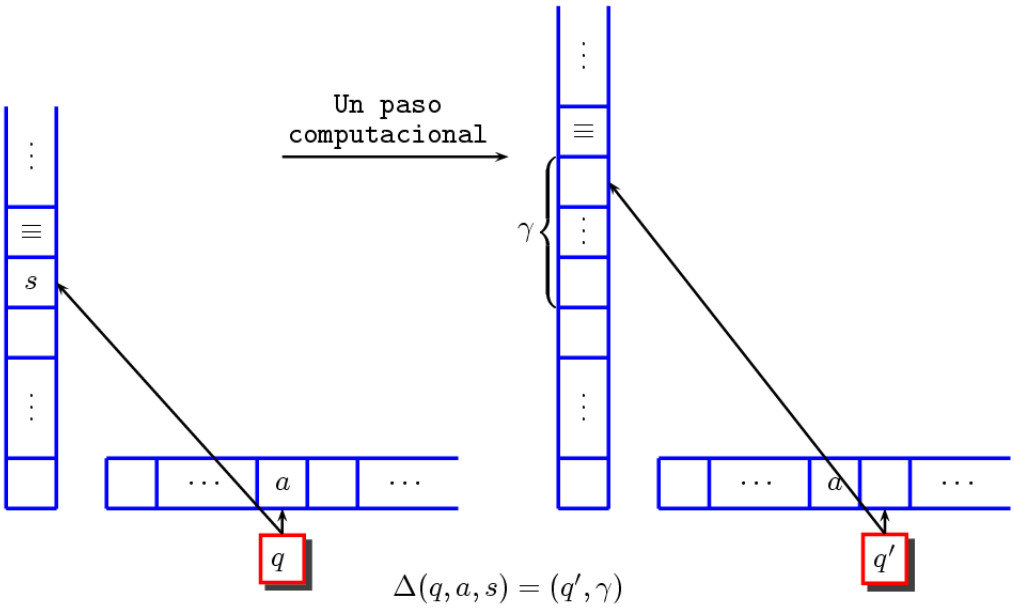
determinado, la unidad de control del autómata escanea un símbolo a sobre la cinta de entrada y el símbolo s en el tope o cima de la pila, como lo muestra la siguiente gráfica:



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 γ (cadena que pertenece a Γ^*) 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.



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 λ o transición espontánea: 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 γ. 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 λ 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, β) a la configuración instantánea (p, v, γ) 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, λ, β), 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 β que queda en la pila puede ser cualquier cadena de símbolos en Γ^*.

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, β) que representa una configuración instantánea.
- La definición de la función de transición Δ 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, λ, 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 λ) 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: *aababbabb* (aceptada).

$$\begin{aligned}(q_0, aababbabb, s_0) &\vdash (q_1, ababbabb, As_0) \vdash (q_1, babbabb, AAs_0)\\ &\vdash (q_1, abbabb, As_0) \vdash (q_1, babb, AAs_0) \vdash (q_1, abb, As_0)\\ &\vdash (q_1, bb, AAs_0) \vdash (q_1, b, As_0) \vdash (q_1, \lambda, s_0) \vdash (q_2, \lambda, s_0).\end{aligned}$$

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

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

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

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 número de ceros que de unos.

5.2. Autómatas con pila no-deterministas (AFPN)

Un **Autómata Finito con Pila No-Determinista** (AFPN) consta de los mismos siete parámetros de un AFPD, $M = (Q, q_0, F, \Sigma, \Gamma, s_0, \Delta)$, pero la función de transición Δ es de la forma:

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

donde $\wp_f(Q \times \Gamma^*)$ es el conjunto de subconjuntos finitos de $Q \times \Gamma^*$. Para $q \in Q$, $a \in \Sigma \cup \{\lambda\}$ y $s \in \Gamma$, $\Delta(q, a, s)$ es de la forma

$$\Delta(q, a, s) = \{(p_1, \gamma_1), (p_2, \gamma_2), \ldots, (p_k, \gamma_k)\}.$$

El significado de esta transición es: al leer el símbolo a sobre la cinta de entrada, la unidad de control puede pasar (aleatoriamente) a uno de los estados p_i y se mueve a la derecha. Sobre la pila hace lo siguiente: borra el símbolo s que está en el tope y escribe la cadena γ_i (cadena que pertenece a Γ^*).

A diferencia de lo que sucede con los AFPD, en el modelo AFPN las transiciones $\lambda, \Delta(q, \lambda, s)$, no tienen restricción alguna.

El lenguaje aceptado por un AFPN M se define como:

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

O sea, una cadena w es aceptada si existe por lo menos un procesamiento de w desde la configuración inicial hasta una configuración de aceptación.

Ejemplo Diseñar un AFPN que acepte el lenguaje $\{a^i b^i : i \geq 0\}$, sobre el alfabeto $\Sigma = \{a, b\}$.

Solución: El lenguaje $\{a^i b^i : i \geq 0\}$ difiere del lenguaje $\{a^i b^i : i \geq 1\}$, utilizado en el primer ejemplo de la sección 5.1, por la presencia de la cadena vacía λ. Para aceptar la cadena λ añadimos una transición más al AFDP diseñado en dicho ejemplo, lo que hace que el autómata sea no-determinista. Concretamente, se define el autómata M como $M = (Q, q_0, F, \Sigma, \Gamma, s_0, \Delta)$ 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:

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

En este autómata el no-determinismo surge por la presencia simultánea de $\Delta(q_0, a, s_0)$ y $\Delta(q_0, \lambda, s_0)$.

Ejemplo Diseñar un AFPN que acepte el lenguaje de todas las cadenas sobre el alfabeto $\Sigma = \{a, b\}$ que tienen igual número de $a$es que de $b$es.

Solución: Simplificamos el autómata del segundo ejemplo de la sección 5.1; sólo se requieren dos estados. Específicamente, $M = (Q, q_0, F, \Sigma, \Gamma, s_0, \Delta)$, donde

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

y la función de transición está dada por:

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

El no-determinismo se presenta únicamente por la presencia simultánea de $\Delta(q_0, a, s_0)$ y $\Delta(q_0, \lambda, s_0)$.

En contraste con lo que sucede con los modelos AFD y AFN, los modelos de autómata con pila determinista (AFPD) y no-determinista (AFPN) no resultan ser computacionalmente equivalentes: existen lenguajes aceptados por AFPD que no pueden ser aceptados por ningún AFPD. Un ejemplo concreto es el lenguaje $L = \{ww^R : w \in \Sigma^*\}$. Como se mostrará a continuación, se puede construir un autómata con pila no-determinista para aceptar a L, pero no es posible diseñar ningún AFPD que lo haga. La

demostración de esta imposibilidad es bastante complicada y no la podemos presentar en el presente libro.

Ejemplo Diseñar un AFPN que acepte el lenguaje $L = \{ww^R : w \in \Sigma^*\}$, donde $\Sigma = \{a, b\}$. No es difícil ver que L es el lenguaje de los palíndromos de longitud par.

Solución: En el último ejemplo de la sección 5.1 se construyó un AFPD que acepta el lenguaje $\{wcw^R : w \in \{a, b\}^*\}$. El lenguaje L del presente ejemplo es similar, excepto que ya no aparece el separador c entre w y w^R. El no-determinismo se puede usar para permitirle al autómata la opción de "adivinar" cuál es la mitad de la cadena de entrada. Si acierta, procederá a comparar el resto de la cadena de entrada con los símbolos acumulados en la pila. Si no acierta, el autómata continuará acumulando símbolos en la pila y no habrá aceptación. Si la cadena de entrada tiene la forma deseada, entre todos los cómputos posibles estará aquél en el que el autómata adivina correctamente cuándo ha llegado a la mitad de la cadena.

M se define como $M = (Q, q_0, F, \Sigma, \Gamma, s_0, \Delta)$ 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:

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

Las dos transiciones

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

le permiten al autómata una opción no-determinista: o seguir acumulando símbolos en la pila, en el estado q_0, o suponer que se ha llegado a la mitad de la cadena de entrada. En este último caso, la unidad de control pasa al estado q_1 y comienza a borrar los símbolos ya almacenados en la pila.

Ejercicios

Diseñar APFN que acepten los siguientes lenguajes:

1. $L = \{a^i b^j c^{i+j} : i, j \geq 0\}$, sobre $\Sigma = \{a, b, c\}$.
2. $L = \{a^{2i} b^{3i} : i, j \geq 0\}$, sobre $\Sigma = \{a, b\}$.
3. $L = \{0^i 1^j : 0 \leq i \leq j \leq 2i\}$, sobre $\Sigma = \{0, 1\}$.
4. $L = \{0^i 1^j : i, j \geq 0, i \neq j\}$, sobre $\Sigma = \{0, 1\}$.

5.3. Aceptación por pila vacía

En todos los modelos de autómatas que hemos considerado en este curso, la aceptación de cadenas está determinada por los estados finales o de aceptación. Para los autómatas con pila existe otra noción de aceptación: la aceptación por pila vacía, definida a continuación. Cuando se usa esta noción, los autómatas no requieren un conjunto F de estados finales, solamente los seis restantes componentes: $(Q, q_0, \Sigma, \Gamma, s_0, \Delta)$.

5.3.1 Definición. Dado un autómata con pila $M = (Q, q_0, \Sigma, \Gamma, s_0, \Delta)$, ya sea AFPD o AFPN, el **lenguaje aceptado por M por pila vacía** se define como

$$N(M) := \{w \in \Sigma^* : \text{existe un cómputo } (q_0, w, s_0) \overset{*}{\vdash} (p, \lambda, \lambda)\}.$$

O sea, una cadena es aceptada por pila vacía si se puede ir, en cero, uno o más pasos, desde la configuración inicial hasta una configuración en la que la pila esté completamente desocupada[1]. Nótese que, para ser aceptada, la cadena de entrada w debe ser procesada completamente.

Para AFPN las nociones de aceptación por pila vacía y por estados finales resultan ser equivalentes, como se establece en los dos siguientes teoremas. Es importante anotar que para autómatas deterministas AFPD los dos tipos de aceptación *no* son equivalentes.

5.3.2 Teorema. *Si $L = L(M)$ para algún autómata con pila AFPN M, entonces $L = N(M')$ para algún AFPN M'. Es decir, M' acepta por pila vacía lo que M acepta por estado final.*

[1] La N en la notación $N(M)$ proviene de la expresión 'pila nula', sinónimo de 'pila vacía'.

Demostración: Sea $M = (Q, q_0, F, \Sigma, \Gamma, s_0, \Delta)$. M' se diseña modificando M de tal manera que vacíe su pila cuando M haya aceptado una cadena de entrada. Concretamente, se define M' como

$$M' = (Q \cup \{p_0, p\}, p_0, \Sigma, \Gamma \cup \{r_0\}, r_0, \Delta')$$

donde p_0 (estado inicial) y p son estados nuevos, y r_0 es el nuevo símbolo inicial de pila. La función de transición Δ' se define así:

1. $\Delta'(p_0, \lambda, r_0) = \{(q_0, s_0 r_0)\}$. Transición λ mediante la cual el nuevo símbolo inicial de pila se coloca en el fondo. Esto impedirá que una cadena sea accidentalmente aceptada si el autómata original M vacía la pila.

2. $\Delta(q, a, s) \subseteq \Delta'(q, a, s)$ para todo $q \in Q$, $a \in \Sigma$ ó $a = \lambda$ y $s \in \Gamma$. Esto quiere decir que M' simula a M: todas las transiciones del autómata original también se pueden realizar en el nuevo autómata.

3. $(p, s) \in \Delta'(q, \lambda, s)$ para todo $q \in F$, $s \in \Gamma \cup \{r_0\}$. Mediante esta transición λ, M' pasa al nuevo estado p siempre que q sea un estado de aceptación.

4. $\Delta'(p, \lambda, s) = \{(p, \lambda)\}$. Mediante esta transición λ, M' borra todo el contenido de la pila.

Obsérvese que las transiciones λ de los numerales 3 y 4 no consumen ningún símbolo en la cadena de entrada. Además, la única manera de que M' vacíe completamente la pila es ingresando al estado p, lo cual puede hacer únicamente desde un estado de aceptación de M.

Si w es aceptada por M, o sea si $w \in L(M)$, M realiza un cómputo de la forma

$$(q_0, w, s_0) \overset{*}{\vdash} (q, \lambda, \beta)$$

donde $q \in F$ y $\beta \in \Gamma^*$. Entonces en M' se puede efectuar el siguiente cómputo:

$$(p_0, w, r_0) \vdash (q_0, w, s_0 r_0) \overset{*}{\vdash} (q, \lambda, \beta r_0) \vdash (p, \lambda, \beta r_0) \overset{*}{\vdash} (p, \lambda, \lambda).$$

Por lo tanto, $w \in N(M')$.

Un razonamiento similar muestra que $w \in N(M')$ implica $w \in L(M)$. En conclusión, $L(M) = N(M')$. □

5.3.3 Teorema. *Si $L = N(M)$ para algún autómata con pila AFPN M, entonces $L = L(M')$ para algún AFPN M'. Es decir, M' acepta por estado final lo que M acepta por pila vacía.*

Demostración: Sea $M = (Q, q_0, \Sigma, \Gamma, s_0, \Delta)$ un AFPN que acepta por pila vacía. M' se diseña añadiendo un nuevo estado q_f a M de tal manera que M' ingrese a tal estado (único estado de aceptación) solamente cuando M haya vaciado su pila. Concretamente, se define M' como

$$M' = (Q \cup \{p_0, p_f\}, p_0, \{p_f\}, \Sigma, \Gamma \cup \{r_0\}, r_0, \Delta')$$

donde p_0 (estado inicial) y p_f (único estado de aceptación) son estados nuevos, y r_0 es el nuevo símbolo inicial de pila. La función de transición Δ' se define así:

1. $\Delta'(p_0, \lambda, r_0) = \{(q_0, s_0 r_0)\}$. Transición λ mediante la cual el nuevo símbolo inicial de pila se coloca en el fondo. Cuando M' encuentre el marcador de fondo r_0, sabrá que M ha vaciado su pila.

2. $\Delta(q, a, s) \subseteq \Delta'(q, a, s)$ para todo $q \in Q$, $a \in \Sigma$ ó $a = \lambda$ y $s \in \Gamma$. Esto quiere decir que M' simula a M: todas las transiciones del autómata original también se pueden realizar en el nuevo autómata.

3. $(p_f, s) \in \Delta'(q, \lambda, r_0)$ para todo $q \in Q$. Mediante esta transición λ, M' pasa al estado de aceptación p_f cuando detecte el marcador de fondo r_0. O sea, M' acepta cuando M vacía su pila.

Si w es aceptada por M, o sea si $w \in N(M)$, M realiza un cómputo de la forma

$$(q_0, w, s_0) \overset{*}{\vdash} (q, \lambda, \lambda)$$

donde $q \in Q$. Entonces en M' se puede efectuar el siguiente cómputo:

$$(p_0, w, r_0) \vdash (q_0, w, s_0 r_0) \overset{*}{\vdash} (q, \lambda, r_0) \vdash (p_f, \lambda, r_0).$$

Por lo tanto, $w \in L(M')$.

Un razonamiento similar muestra que $w \in L(M')$ implica $w \in N(M)$. En conclusión, $N(M) = L(M')$. □

Ejercicios

1. Modificar los autómatas de los tres ejemplos de la sección 5.2 para que acepten por pila vacía y no por estado final.

2. Diseñar AFPN que acepten los siguientes lenguajes por pila vacía:

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

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

(III) $L = \{0^i 1^j : 0 \leq i \leq j \leq 2i\}$, sobre $\Sigma = \{0, 1\}$.

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

3. Completar los detalles faltantes en las demostraciones del Teorema 5.3.2 y el Teorema 5.3.3. ¿Por qué estas demostraciones no son válidas para autómatas deterministas?

5.4. Autómatas con pila y LIC (Parte I)

Los lenguajes aceptados por los AFPN son exactamente los lenguajes independientes del contexto. Éste es un resultado análogo al Teorema de Kleene para lenguajes regulares, aunque en el caso de los autómatas con pila, los modelos deterministas no son computacionalmente equivalentes a los no-deterministas. En la presente sección consideraremos la primera parte de la correspondencia entre AFPN y LIC.

5.4.1 Teorema. *Dada una GIC G, existe un AFPN M tal que $L(G) = L(M)$.*

Bosquejo de la demostración. Para una gramática $G = (\Sigma, V, S, P)$ dada, se construye un AFPN que utiliza la pila para simular la derivación de cadenas realizada por G. M requiere solamente tres estados, independientemente del número de variables y producciones de G. Específicamente, el autómata M se define como $M = (Q, q_0, F, \Sigma, \Gamma, s_0, \Delta)$, donde $Q = \{q_0, q_1, q_2\}$, $F = \{q_2\}$ y $\Gamma = \Sigma \cup V \cup \{s_0\}$. La función de transición Δ se define de la siguiente manera:

1. $\Delta(q_0, \lambda, s_0) = \{(q_1, Ss_0)\}$. Transición λ mediante la cual M coloca el símbolo S en el tope de la pila al iniciar el procesamiento de una cadena de entrada.

2. Para cada variable $A \in V$,

$$\Delta(q_1, \lambda, A) = \{(q_1, u) : A \to u \text{ es una producción de la gramática } G\}.$$

   Mediante estas transiciones, M utiliza la pila para simular las derivaciones: si el tope de la pila es A y en la derivación se usa la producción $A \to u$, el tope de la pila A es substituido por u.

3. Para cada símbolo terminal $a \in \Sigma$, $\Delta(q_1, a, a) = \{(q_1, \lambda)\}$. Mediante estas transiciones, M borra los terminales del tope de la pila al consumirlos sobre la cinta de entrada.

4. $\Delta(q_1, \lambda, s_0) = \{(q_2, s_0)\}$. M ingresa al estado de aceptación q_2 cuando detecta el marcador de fondo s_0.

El autómata M está diseñado de tal forma que si $S \overset{*}{\Longrightarrow} w$ es una derivación a izquierda en la gramática G, entonces existe un procesamiento

$$(q_0, w, s_0) \vdash (q_0, w, Ss_0) \overset{*}{\vdash} (q_2, \lambda, s_0)$$

que simula la derivación.

Recíprocamente, puede demostrarse que si $(q_0, w, s_0) \overset{*}{\vdash} (q_2, \lambda, s_0)$ entonces $S \overset{*}{\Longrightarrow} w$ en la gramática G. □

Ejemplo Sea G la gramática:

$$G : \begin{cases} S \to aAbS \mid bBa \mid \lambda \\ A \to aA \mid a \\ B \to bB \mid b \end{cases}$$

Según la construcción del Teorema 5.4.1, el autómata M está dado por $M = (Q, q_0, F, \Sigma, \Gamma, s_0, \Delta)$ donde

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

La función de transición Δ está dada por:

$$\begin{aligned} \Delta(q_0, \lambda, s_0) &= \{(q_1, Ss_0)\}, \\ \Delta(q_1, \lambda, S) &= \{(q_1, aAbS), (q_1, bBa), (q_1, \lambda)\}, \\ \Delta(q_1, \lambda, A) &= \{(q_1, aA), (q_1, a)\}, \\ \Delta(q_1, \lambda, B) &= \{(q_1, bB), (q_1, b)\}, \\ \Delta(q_1, a, a) &= \{(q_1, \lambda)\}, \\ \Delta(q_1, b, b) &= \{(q_1, \lambda)\}, \\ \Delta(q_1, \lambda, s_0) &= \{(q_2, s_0)\}. \end{aligned}$$

Podemos ilustrar la correspondencia entre derivaciones en G y procesamientos en M con la cadena $aabbba$, la cual tiene la siguiente derivación a izquierda:

$$S \Longrightarrow aAbS \Longrightarrow aabS \Longrightarrow aabbBa \Longrightarrow aabbba.$$

El autómata M simula esta derivación de la cadena $aabbba$ así:

$$\begin{aligned} (q_0, aabbba, s_0) &\vdash (q_1, aabbba, Ss_0) \vdash (q_1, aabbba, aAbSs_0) \\ &\vdash (q_1, abbba, AbSs_0) \vdash (q_1, abbba, abSs_0) \\ &\vdash (q_1, bbba, bSs_0) \vdash (q_1, bba, Ss_0) \vdash (q_1, bba, bBas_0) \\ &\vdash (q_1, ba, Bas_0) \vdash (q_1, ba, bas_0) \vdash (q_1, a, as_0) \\ &\vdash (q_1, \lambda, s_0) \vdash (q_2, \lambda, s_0). \end{aligned}$$

Ejemplo La siguiente gramática genera los palíndromos de longitud par, sobre $\Sigma = \{a, b\}$, es decir, el lenguaje $L = \{ww^R : w \in \Sigma^*\}$:

$$S \to aSa \mid bSb \mid \lambda.$$

Siguiendo el procedimiento del Teorema 5.4.1 podemos construir un AFPN que acepta a L; este autómata es diferente del exhibido en el tercer ejemplo de la sección 5.2. $M = (Q, q_0, F, \Sigma, \Gamma, s_0, \Delta)$ donde

$$\begin{aligned} Q &= \{q_0, q_1, q_2\}, \\ F &= \{q_2\}, \\ \Gamma &= \{a, b, S, s_0\}. \end{aligned}$$

La función de transición Δ está dada por:

$$\begin{aligned} \Delta(q_0, \lambda, s_0) &= \{(q_1, Ss_0)\}, \\ \Delta(q_1, \lambda, S) &= \{(q_1, aSa), (q_1, bSb), (q_1, \lambda)\}, \\ \Delta(q_1, a, a) &= \{(q_1, \lambda)\}, \\ \Delta(q_1, b, b) &= \{(q_1, \lambda)\}, \\ \Delta(q_1, \lambda, s_0) &= \{(q_2, s_0)\}. \end{aligned}$$

Ejercicios

1. Construir un AFPN M que acepte el lenguaje generado por la siguiente gramática:

$$G: \quad \begin{cases} S \to Aba \mid AB \mid \lambda \\ A \to aAS \mid a \\ B \to bBA \mid \lambda. \end{cases}$$

   Encontrar una derivación a izquierda en G de la cadena $w = aaababa$ y procesar luego la cadena w con el autómata M, simulando la derivación.

2. Construir un AFPN M que acepte el lenguaje generado por la siguiente gramática:

$$G: \quad \begin{cases} S \to ASA \mid AaA \mid aa \\ A \to AbA \mid \lambda. \end{cases}$$

   Encontrar una derivación a izquierda en G de la cadena $w = bbaaa$ y procesar luego la cadena w con el autómata M, simulando la derivación.

5.5. Autómatas con pila y LIC (Parte II)

En la presente sección consideraremos la segunda parte de la correspondencia entre AFPN y LIC. Demostraremos que para todo AFPN que acepta por pila vacía existe una GIC que genera el lenguaje aceptado por el autómata. Las gramáticas obtenidas son, en general, bastante complejas, con un gran número de variables y de producciones. Hay que advertir también que el procedimiento puede dar lugar a variables inútiles (no-terminables o no alcanzables).

5.5.1 Teorema. *Dado un AFPN* $M = (Q, q_0, \Sigma, \Gamma, s_0, \Delta)$ *que acepta por pila vacía, existe una GIC* $G = (\Sigma, V, S, P)$ *tal que* $L(G) = N(M)$.

Demostración: En la gramática G las variables (aparte de la variable inicial S) serán tripletas de la forma $[qXp]$ donde q, $p \in Q$ y $X \in \Gamma$. Las producciones de G se definen de la siguiente manera:

1. Si $(p, \lambda) \in \Delta(q, a, X)$, se añade la producción $[qXp] \to a$.
2. Si $(p, \lambda) \in \Delta(q, \lambda, X)$, se añade la producción $[qXp] \to \lambda$.
3. Si $(r, Y_1Y_2 \cdots Y_k) \in \Delta(q, a, X)$, donde a puede ser un símbolo del alfabeto Σ ó $a = \lambda$ y $k \geq 1$, se añaden todas las producciones de la forma
$$[qXr_k] \to a[rY_1r_1][r_1Y_2r_2] \cdots [r_{k-1}Y_kr_k]$$
para todas las secuencias posibles r_1, r_2, $\ldots$, r_{k-1} de estados de Q.
4. Para todo $p \in Q$ se añade la producción $S \to [q_0s_0p]$.

La gramática G así definida pretende simular con derivaciones a izquierda los cómputos de M; el significado intuitivo de la variable $[qXp]$ es: "al extraer X del tope de la pila, se pasa del estado q al estado p". La producción
$$[qXr_k] \to a[rY_1r_1][r_1Y_2r_2] \cdots [r_{k-1}Y_kr_k]$$
del numeral 3 indica las posibles maneras en las que M puede extraer la cadena $Y_1Y_2 \cdots Y_k$ de la pila, una vez se haya sustituido el tope de la pila X por dicha cadena, pasando del estado q al estado r y consumiendo el símbolo a.

Demostraremos primero la inclusión $N(M) \subseteq L(G)$. Para todo q, $p \in Q$, $X \in \Gamma$ y $w \in \Sigma^*$, se demostrará la implicación

$$\text{si } (q, w, X) \overset{+}{\vdash} (p, \lambda, \lambda) \text{ entonces } [qXp] \overset{+}{\Longrightarrow} w, \tag{5.1}$$

por inducción sobre el número de pasos del cómputo $(q, w, X) \overset{+}{\vdash} (p, \lambda, \lambda)$. Cuando hay un sólo paso, el cómputo es de la forma $(q, a, X) \vdash (p, \lambda, \lambda)$ o de la forma $(q, \lambda, X) \vdash (p, \lambda, \lambda)$. Si $(q, a, X) \vdash (p, \lambda, \lambda)$, entonces $(p, \lambda) \in \Delta(q, a, X)$; así que

$[qXp] \to a$ es una producción de G, y se obtendrá la derivación $[qXp] \Longrightarrow a$. Si $(q, \lambda, X) \vdash (p, \lambda, \lambda)$, entonces $(p, \lambda) \in \Delta(q, \lambda, X)$; así que $[qXp] \to \lambda$ es una producción de G y se obtendrá $[qXp] \Longrightarrow \lambda$.

Para el razonamiento inductivo, supóngase que $(q, w, X) \overset{n}{\vdash} (p, \lambda, \lambda)$ donde $n > 1$. Considerando el primer paso de este cómputo de n pasos, podemos escribir:

$$(q, ax, X) \vdash (r_0, x, Y_1Y_2 \cdots Y_k) \overset{*}{\vdash} (p, \lambda, \lambda), \tag{5.2}$$

donde $a \in \Sigma$ ó $a = \lambda$. Cuando $a \in \Sigma$, $w = ax$ para alguna cadena $x \in \Sigma^*$; cuando $a = \lambda$, $x = w$. En el primer paso de 5.2 se ha aplicado la transición $(r_0, Y_1Y_2 \cdots Y_k) \in \Delta(q, a, X)$ de M. Por la definición de la gramática G,

$$[qXr_k] \to a[r_0Y_1r_1][r_1Y_2r_2] \cdots [r_{k-1}Y_kr_k]$$

es una producción, para todas las secuencias posibles r_1, r_2, ..., r_{k-1} de estados de Q.

Según 5.2, desde la configuración instantánea $(r_0, x, Y_1Y_2 \cdots Y_k)$ el autómata llega hasta la configuración (p, λ, λ), consumiendo completamente la cadena x y vaciando la pila. La cadena x se puede escribir entonces como $x = w_1w_2 \cdots w_k$, siendo w_i la cadena consumida por el autómata para extraer el símbolo Y_i del tope de la pila. En consecuencia, existe una secuencia de estados r_1, r_2,..., r_{k-1}, $r_k = p$ tales que

$$\begin{aligned}(r_0, x, Y_1Y_2 \cdots Y_k) = (r_0, w_1w_2 \cdots w_k, Y_1Y_2 \cdots Y_k) &\overset{+}{\vdash} (r_1, w_2 \cdots w_k, Y_2 \cdots Y_k) \\ &\overset{+}{\vdash} (r_2, w_3 \cdots w_k, Y_3 \cdots Y_k) \overset{+}{\vdash} (r_{k-1}, w_k, Y_k) \overset{+}{\vdash} (r_k, \lambda, \lambda).\end{aligned}$$

Para $i = 1, 2, \ldots, k$ se tiene así una secuencia de cómputos parciales

$$(r_{i-1}, w_i, Y_i) \overset{+}{\vdash} (r_i, \lambda, \lambda),$$

donde $r_k = p$. Por la hipótesis de inducción, $[r_{i-1}Y_ir_i] \overset{+}{\Longrightarrow} w_i$ para $i = 1, 2, \ldots, k$. Por consiguiente,

$$[qXp] = [qXr_k] \Longrightarrow a[r_0Y_1r_1][r_1Y_2r_2] \cdots [r_{k-1}Y_kr_k] \overset{+}{\Longrightarrow} aw_1w_2 \cdots w_k = w.$$

Esto demuestra la implicación 5.1. Por lo tanto, si w es aceptada por M, siendo $w \neq \lambda$, se tendrá $(q_0, w, s_0) \overset{+}{\vdash} (p, \lambda, \lambda)$, y usando 5.1 se concluirá que $S \Longrightarrow [q_0s_0p] \overset{+}{\Longrightarrow} w$. Si $w = \lambda$ es aceptada por M, necesariamente $(q_0, \lambda, s_0) \vdash (p, \lambda, \lambda)$ para algún estado p. Esto significa que $(q_0, \lambda, s_0) \vdash (p, \lambda, \lambda)$, y se tendrá $S \Longrightarrow [q_0s_0p] \Longrightarrow \lambda$. Esto demuestra que $N(M) \subseteq L(G)$.

Para establecer la contenencia $L(G) \subseteq N(M)$ se demuestra la implicación

$$\text{si } [qXp] \overset{+}{\Longrightarrow} w, \text{ entonces } (q, w, X) \overset{+}{\vdash} (p, \lambda, \lambda), \tag{5.3}$$

por inducción sobre el número de pasos en la derivación $[qXp] \overset{+}{\Longrightarrow} w$. Este razonamiento inductivo es similar al usado para probar la implicación recíproca 5.1, y se deja como ejercicio para el lector interesado. Si en G se puede derivar la cadena w, es decir, si $S \overset{*}{\Longrightarrow} w$, la primera producción aplicada será de la forma $S \to [q_0 s_0 p]$. De donde, $S \Longrightarrow [q_0 s_0 p] \overset{+}{\Longrightarrow} w$. Usando 5.3 se concluye $(q_0, w, s_0) \overset{+}{\vdash} (p, \lambda, \lambda)$, o sea $w \in N(M)$. □

Ejemplo Vamos a aplicar la construcción del Teorema 5.5.1 para encontrar una gramática G que genere el lenguaje de las cadenas sobre el alfabeto $\Sigma = \{a, b\}$ que tienen igual número de $a$es que de $b$es, a partir del AFPN $M = (Q, q_0, F, \Sigma, \Gamma, s_0, \Delta)$ con los siguientes componentes. $\Sigma = \{a, b\}$, $\Gamma = \{s_0, A, B\}$, $Q = \{q_0, q_1\}$, $F = \{q_1\}$ y la función de transición Δ está dada por:

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

M es una modificación del autómata presentado en el segundo ejemplo de la sección 5.2 (en ese ejemplo M aceptaba por estado final; aquí M acepta por pila vacía).

Las variables de G son S y todas las tripletas de la forma $[qXp]$ donde q, $p \in Q$ y $X \in \Gamma$. Hay, por lo tanto, 13 variables, a saber:

$$\begin{aligned}
&S,\ [q_0 s_0 q_0],\ [q_0 A q_0],\ [q_0 B q_0],\ [q_0 s_0 q_1],\ [q_0 A q_1],\ [q_0 B q_1],\\
&[q_1 s_0 q_0],\ [q_1 A q_0],\ [q_1 B q_0],\ [q_1 s_0 q_1],\ [q_1 A q_1],\ [q_1 B q_1].
\end{aligned}$$

A continuación se presentan las producciones de G.

Producción obtenida de $\Delta(q_0, \lambda, s_0) = \{(q_1, \lambda)\}$:

$$[q_0 s_0 q_1] \to \lambda.$$

Producción obtenida de $\Delta(q_0, a, B) = \{(q_0, \lambda)\}$:

$$[q_0 B q_0] \to a.$$

Producción obtenida de $\Delta(q_0, b, A) = \{(q_0, \lambda)\}$:

$$[q_0 A q_0] \to b.$$

Producciones obtenidas de $\Delta(q_0, a, s_0) = \{(q_0, As_0)\}$:

$$[q_0 s_0 q_0] \to a[q_0 A q_0][q_0 s_0 q_0] \mid a[q_0 A q_1][q_1 s_0 q_0]$$
$$[q_0 s_0 q_1] \to a[q_0 A q_0][q_0 s_0 q_1] \mid a[q_0 A q_1][q_1 s_0 q_1].$$

Producciones obtenidas de $\Delta(q_0, b, s_0) = \{(q_0, Bs_0)\}$:

$$[q_0 s_0 q_0] \to b[q_0 B q_0][q_0 s_0 q_0] \mid b[q_0 B q_1][q_1 s_0 q_0]$$
$$[q_0 s_0 q_1] \to b[q_0 B q_0][q_0 s_0 q_1] \mid b[q_0 B q_1][q_1 s_0 q_1].$$

Producciones obtenidas de $\Delta(q_0, a, A) = \{(q_0, AA)\}$:

$$[q_0 A q_0] \to a[q_0 A q_0][q_0 A q_0] \mid a[q_0 A q_1][q_1 A q_0]$$
$$[q_0 A q_1] \to a[q_0 A q_0][q_0 A q_1] \mid a[q_0 A q_1][q_1 A q_1].$$

Producciones obtenidas de $\Delta(q_0, b, B) = \{(q_0, BB)\}$:

$$[q_0 B q_0] \to b[q_0 B q_0][q_0 B q_0] \mid b[q_0 B q_1][q_1 B q_0]$$
$$[q_0 B q_1] \to b[q_0 B q_0][q_0 B q_1] \mid b[q_0 B q_1][q_1 B q_1].$$

Finalmente, las producciones de la variable inicial S son:

$$S \to [q_0 s_0 q_0] \mid [q_0 s_0 q_1].$$

Al examinar las producciones se puede observar que todas las variables de la forma $[q_1 X q]$, con $X \in \Gamma$ y $q \in Q$, son inútiles ya que no tienen producciones. Con la terminología ya conocida, dichas variables son no-terminables y, por consiguiente, las producciones en las que aparecen se pueden eliminar. Otra variable no terminable es $[q_0 s_0 q_0]$. Además, las variables $[q_0 A q_1]$ y $[q_0 B q_1]$ son inalcanzables, así que se pueden eliminar, junto con todas sus producciones. Realizando estas simplificaciones se obtiene la siguiente gramática:

$$S \to [q_0 s_0 q_1]$$
$$[q_0 s_0 q_1] \to a[q_0 A q_0][q_0 s_0 q_1] \mid b[q_0 B q_0][q_0 s_0 q_1] \mid \lambda$$
$$[q_0 A q_0] \to a[q_0 A q_0][q_0 A q_0] \mid b$$
$$[q_0 B q_0] \to b[q_0 B q_0][q_0 B q_0] \mid a.$$

Como se indicó en la demostración, en las gramáticas construidas según el método del Teorema 5.5.1, las derivaciones a izquierda corresponden a los cómputos en el autómata

dado. Podemos ilustrar este punto, en el presente ejemplo, con la cadena de entrada $w = bbabaa$. El siguiente es un cómputo de aceptación de w en M:

$$\begin{aligned}(q_0, bbabaa, s_0) &\vdash (q_0, babaa, Bs_0) \vdash (q_0, abaa, BBs_0) \vdash (q_0, baa, Bs_0)\\ &\vdash (q_0, aa, BBs_0) \vdash (q_0, a, Bs_0) \vdash (q_0, \lambda, s_0) \vdash (q_1, \lambda, \lambda).\end{aligned}$$

La derivación a izquierda en G que corresponde a este cómputo es:

$$\begin{aligned}S &\Longrightarrow [q_0s_0q_1] \Longrightarrow b[q_0Bq_0][q_0s_0q_1] \Longrightarrow bb[q_0Bq_0][q_0Bq_0][q_0s_0q_1]\\ &\Longrightarrow bba[q_0Bq_0][q_0s_0q_1] \Longrightarrow bbab[q_0Bq_0][q_0Bq_0][q_0s_0q_1]\\ &\Longrightarrow bbaba[q_0Bq_0][q_0s_0q_1] \Longrightarrow bbabaa[q_0s_0q_1] \Longrightarrow bbabaa.\end{aligned}$$

Obsérvese que, en cada paso de la derivación, el contenido actual de la pila se puede leer examinando las segundas componentes de las tripletas.

Para hacer más legibles las producciones de la gramática G obtenida en este ejemplo, cambiamos los nombres de las variables así: $C = [q_0s_0q_1]$, $D = [q_0Aq_0]$ y $E = [q_0Bq_0]$. Con esta nomenclatura, la gramática se puede escribir como:

$$G: \quad \begin{cases} S \to C \\ C \to aDC \mid bEC \mid \lambda \\ D \to aDD \mid b \\ E \to bEE \mid a. \end{cases}$$

Puesto que la única producción de S es $S \to C$, las variables S y C se pueden identificar, dando lugar a la siguiente gramática simplificada equivalente:

$$\begin{cases} S \to aDS \mid bES \mid \lambda \\ D \to aDD \mid b \\ E \to bEE \mid a. \end{cases}$$

Ejercicios

1. Con respecto al ejemplo de esta sección, procesar con M la cadena de entrada $w = baabbbaa$ y luego derivar w en la gramática G, simulando dicho procesamiento.

2. Modificar el AFPN presentado en el último ejemplo de la sección 5.2 para que acepte por pila vacía el lenguaje $L = \{ww^R : w \in \{a, b\}^*\}$. Aplicar luego la construcción del Teorema 5.5.1 para encontrar una gramática cuyo lenguaje generado sea L.