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:

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

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 las cadenas sobre el alfabeto $\Sigma = \{a, b\}$ que tiene 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.

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:

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

Las dos transiciones

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

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 de la sección 5.2

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 $\Sigma = \{0, 1\}$.