1.5. Autómatas finitos no deterministas

Definición. Un *autómata finito no determinista (AFND)* es una 5-tupla $M = (Q, \Sigma, \delta, q_0, F)$, donde

- Q es un conjunto finito de *estados*;
- Σ es un conjunto finito llamado *alfabeto*;
- $\delta\colon Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$ es la *función de transición*;
- $q_0 \in Q$ es el *estado inicial*; y
- $F \subseteq Q$ es el conjunto de *estados de aceptación (o finales)*.

Definición. Sea $M = (Q, \Sigma, \delta, q_0, F)$ un AFND y sea $w \in \Sigma^*$. Diremos que *M acepta w* si, para algún $m \geq 0$, podemos escribir

$$w = x_1 x_2 \cdots x_m, \qquad x_i \in \Sigma \cup \{\varepsilon\},$$

y existen estados $r_0, r_1, \ldots, r_m \in Q$ tales que

- $r_0 = q_0$;
- $r_i \in \delta(r_{i-1}, x_i)$ para todo $i \in \{1, \ldots, m\}$; y
- $r_m \in F$.

Si M acepta w, escribimos $M(w) = 1$; de lo contrario, escribimos $M(w) = 0$.

Definición. Sea $M = (Q, \Sigma, \delta, q_0, F)$ un AFND. El *lenguaje aceptado por M* es

$$\mathcal{L}(M) = \{w \in \Sigma^* : M(w) = 1\}.$$

1.6. Equivalencia entre AFND y AFD

Definición. Dos máquinas A y B son *equivalentes* si

$$\mathcal{L}(A) = \mathcal{L}(B).$$

Teorema. Todo AFND tiene un AFD equivalente.

Corolario. Sea $L \subseteq \Sigma^*$ un lenguaje. El lenguaje L es regular si y sólo si existe un AFND M tal que $\mathcal{L}(M) = L$.

Ejercicios

1. *a*) Construya un AFND que acepte el lenguaje

$$\{w \in \{0,1\}^* : w \text{ contiene 001 como substring o termina en 110}\}.$$

b) Considere el lenguaje

$$F = \{w \in \{0,1\}^* : w \text{ no contiene un par de 1's separados por una cantidad impar de caracteres}\}$$

Muestre que F es regular.
Indicación: estudie F^c.

c) Considere el AFND sin transiciones ε con estado inicial p, único estado final r y tabla

	a	b
p	$\{p,q\}$	$\{p\}$
q	$\emptyset$	$\{r\}$
r	$\{r\}$	$\{r\}$

.

Construya un AFD equivalente y dibuje ambos autómatas. Describa además el lenguaje aceptado por ambos autómatas.

d) Para complementar el lenguaje de un AFD basta intercambiar sus estados finales y no finales. ¿Funciona siempre la misma operación para un AFND? Demuestre o entregue un contraejemplo.

e) Dado $w = w_1 \dots w_n \in \Sigma^*$ definimos su *reverso* como $w^r = w_n \dots w_1$. Para un lenguaje $L \subseteq \Sigma^*$, definimos

$$L^r = \{w^r \in \Sigma^* : w \in L\}.$$

Muestre que si L es regular, entonces L^r también es regular.

f) Sean A y B dos lenguajes regulares. Muestre que

$$\text{EVITA}(A,B) = \{a \in A : a \text{ no contiene a ningún } b \in B\}$$

es regular.

2. Dada $w \in \{0,1,?\}^*$, diremos que x es una *completación* de w si x se obtiene de w reemplazando cada símbolo ? por un 0 o un 1.

   a) Dado un lenguaje $L \subseteq \{0,1\}^*$, definimos

   $$\text{COMP}_\exists(L) = \{w \in \{0,1,?\}^* : \text{existe una completación de } w \text{ en } L\}.$$

   Sea

   $$L_0 = \{w \in \{0,1\}^* : w \text{ contiene 01 como substring}\}.$$

   Muestre que $\text{COMP}_\exists(L_0)$ es regular.

   b) Muestre que si L es regular, entonces $\text{COMP}_\exists(L)$ también es regular.

   c) Definimos ahora

   $$\text{COMP}_\forall(L) = \{w \in \{0,1,?\}^* : \text{todas las completaciones de } w \text{ están en } L\}.$$

   Muestre que si L es regular, entonces $\text{COMP}_\forall(L)$ también es regular.

   d) Definimos finalmente

   $$\text{COMP}_{\text{mix}}(L) = \{w \in \{0,1,?\}^* : \text{existe una completación de } w \text{ en } L \text{ y una en } L^c\}.$$

   Muestre que si L es regular, entonces $\text{COMP}_{\text{mix}}(L)$ también es regular.

3. Para cada $n \geq 1$, considere el lenguaje

   $$L_n = \{w \in \{0,1\}^* : |w| \geq n \text{ y el } n\text{-ésimo símbolo de } w, \text{ contado desde la derecha, es } 1\}.$$

   a) Construya un AFND con $n+1$ estados que reconozca L_n.

   b) Sea $L \subseteq \Sigma^*$. Una *familia engañadora* para L es una colección de pares

   $$\{(x_i, y_i) : 1 \leq i \leq m\}$$

   tal que $x_i y_i \in L$ para todo i y, para cada $i \neq j$, al menos uno de los strings $x_i y_j$ y $x_j y_i$ no pertenece a L. Demuestre que, si L posee una familia engañadora de tamaño m, todo AFND que reconoce L tiene al menos m estados.

   c) Construya una familia engañadora de tamaño $n+1$ para L_n. Concluya que el AFND construido en el inciso (a) es mínimo.

   d) El ejercicio de la clase anterior vimos que el AFD mínimo que reconoce L_n tiene 2^n estados. En esta pregunta concluimos que el ANFD mínimo que reconoce L_n tiene $n+1$ estados. ¿Qué tan grande puede ser la brecha entre el número de estados del AFD mínimo y el número de estados del AFND mínimo que reconocen un mismo lenguaje regular?