

0.1. Autómatas finitos deterministas

Definición. Un *autómata finito determinista (AFD)* 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*
- $\delta: Q \times \Sigma \rightarrow Q$ es la *función de transición*.
- $q_0 \in Q$ es el *estado inicial*, y
- $F \subseteq Q$ son los *estados de aceptación (o finales)*.

Definición. Sea $M = (Q, \Sigma, \delta, q_0, F)$ un AFD y $w = w_1 \dots w_n \in \Sigma^*$ un string. Diremos que M *acepta* w si existen estados $r_0, r_1, \dots, r_n$ tal que

- $r_0 = q_0$,
- $r_i = \delta(r_{i-1}, w_i)$ para todo $i \in \{1, \dots, n\}$,
- $r_n \in F$.

Si M acepta w , usaremos también la notación $M(w) = 1$; de lo contrario, usaremos la notación $M(w) = 0$.

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

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

Definición. Sea Σ un alfabeto y $L_1, L_2 \subseteq \Sigma^*$ dos lenguajes regulares. Los siguientes son lenguajes regulares

- L_1^c
- $L_1 \cup L_2$
- $L_1 \cap L_2$

Ejercicios

1. a) Pruebe que el siguiente lenguaje es regular:

$$L_1 = \{w \in \{0, 1\}^* : |w|_1 \equiv 2 \pmod{3}\},$$

donde $|w|_1$ denota la cantidad de unos en w .

- b) Dado un string $w \in \{0, 1\}^*$, definimos $\text{val}_2(w)$ como el valor binario de w . Más formalmente, definimos recursivamente

$$\text{val}_2(\varepsilon) = 0 \quad \text{y} \quad \text{val}_2(wb) = 2 \text{val}_2(w) + b$$

para $w \in \{0, 1\}^*$ y $b \in \{0, 1\}$. Muestre que el siguiente lenguaje es regular:

$$L_2 = \{w \in \{0, 1\}^* : \text{val}_2(w) \equiv 0 \pmod{5}\}.$$

- c) Muestre que el siguiente lenguaje es regular:

$$L_3 = \{w \in \{0, 1\}^* : \text{val}_2(w) \not\equiv 0 \pmod{5} \text{ y } |w|_1 \equiv 2 \pmod{3}\}.$$

- d) Sean $A, B \subseteq \Sigma^*$ dos lenguajes sobre el mismo alfabeto. Definimos su *barajado perfecto* como

$$\text{shuffle}(A, B) = \{a_1 b_1 a_2 b_2 \cdots a_n b_n : n \geq 0, a_1 \cdots a_n \in A \text{ y } b_1 \cdots b_n \in B\}.$$

Demuestre que los lenguajes regulares son cerrados bajo barajado perfecto.

2. Para cada $n \geq 1$, considere el lenguaje sobre $\Sigma = \{0, 1\}$ definido como

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

El objetivo de esta pregunta es determinar la cantidad óptima de estados que necesita un AFD para decidir L_n .

- Construya un AFD con 8 estados que acepte L_3 . ¿Qué información guarda su automata en los estados?
- Construya un AFD con 2^n estados que acepte L_n . ¿Qué información guarda su automata en los estados?
- Sea M cualquier AFD y $x, y \in \Sigma^*$. Demuestre que, si M queda en el mismo estado después de leer x y después de leer y , entonces $M(xz) = M(yz)$ para todo $z \in \Sigma^*$.
- Suponga ahora que M reconoce L_n . Demuestre que dos strings distintos $x, y \in \{0, 1\}^n$ deben llevar a M a estados distintos. Concluya que cualquier AFD que reconozca L_n debe tener al menos 2^n estados.
- Discuta que consecuencias tiene el inciso anterior sobre su automata del inciso b). ¿Qué dice esto sobre la cantidad de bits de memoria que se necesitan para aceptar L_n ?

3. Sea $M = (Q, \Sigma, \delta, q_0, F)$ un AFD. Definimos la *función de transición extendida* $\widehat{\delta}: Q \times \Sigma^* \rightarrow Q$ recursivamente como

$$\widehat{\delta}(q, \varepsilon) = q \quad \text{y} \quad \widehat{\delta}(q, xa) = \delta(\widehat{\delta}(q, x), a).$$

Para $S \subseteq Q$ no vacío y $w \in \Sigma^*$, defina además

$$\widehat{\delta}(S, w) = \{\widehat{\delta}(q, w) : q \in S\}.$$

- a) Considere el AFD $A = (Q, \Sigma, \delta, q_0, F)$ con $Q = \{0, 1, 2\}$, $\Sigma = \{a, b\}$, $q_0 = 0$, $F = \{0\}$ y tabla de transición

	a	b
0	1	0
1	2	1
2	0	0

Calcule $\widehat{\delta}(Q, \varepsilon)$, $\widehat{\delta}(Q, b)$, $\widehat{\delta}(Q, ba)$, $\widehat{\delta}(Q, baa)$, $\widehat{\delta}(Q, baab)$.

- b) Un string $w \in \Sigma^*$ es *sincronizante* para M si llega al mismo estado independientemente del estado inicial; es decir, si $|\widehat{\delta}(Q, w)| = 1$. Un AFD que posee al menos un string sincronizante se llama *sincronizable*. ¿Es el AFD del inciso anterior sincronizable?
- c) Demuestre que, si w es sincronizante, entonces xwy también lo es para todos $x, y \in \Sigma^*$.
- d) Defina

$$\text{Sync}(M) = \{w \in \Sigma^* : w \text{ es sincronizante para } M\}.$$

Muestre que $\text{Sync}(M)$ es un lenguaje regular.

- e) Deduzca de la parte anterior (1) un algoritmo que decida si un AFD es sincronizable y (2) que si un AFD con n estados es sincronizable, entonces existe un string sincronizante de largo $2^n - n - 1$.
- f) ¿Todo AFD es sincronizable? Demuéstrelo o entregue un contraejemplo.