3.1. Máquinas de Turing

Definición. Una *máquina de Turing* es una 7-tupla $(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ donde Q, Σ, Γ son conjuntos finitos y

1. Q es el conjunto de estados,
2. Σ es el alfabeto de entrada, que no contiene el símbolo blanco $\sqcup$,
3. Γ es el alfabeto de cinta, donde $\sqcup \in \Gamma$ y $\Sigma \subseteq \Gamma$,
4. $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ es la función de transición,
5. $q_0 \in Q$ es el estado inicial,
6. $q_a \in Q$ es el estado de aceptación, y
7. $q_r \in Q$ es el estado de rechazo, donde $q_r \neq q_a$.

Definición. Sea M una MT. Una *configuración* es un string $\alpha\ q\ \beta \in \Gamma^* Q \Gamma^*$ donde α son los contenidos de la cinta a la izquierda del cabezal, q es el estado actual de la máquina, y β es el resto de la cinta.

Definición. Sea $M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ una MT. Sea $u, v \in \Gamma^*$, $a, b, c \in \Gamma$, $q \in Q \setminus \{q_a, q_r\}$ y $r \in Q$. Diremos que M *computa en un paso* desde $ua\ q\ bv$:

- La configuración $uac\ r\ v$ si $\delta(q, b) = (r, c, R)$.
- La configuración $u\ r\ acv$ si $\delta(q, b) = (r, c, L)$.

Si M computa desde la config. C la config. D en un paso, lo denotaremos por $C \vdash_M D$.

Definición (Caso borde)**.** Sea $M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ una MT. Sea $v \in \Gamma^*$, $b, c \in \Gamma$, $q \in Q \setminus \{q_a, q_r\}$ y $r \in Q$. Diremos que M *computa en un paso* desde $q\ bv$:

- La configuración $c\ r\ v$ si $\delta(q, b) = (r, c, R)$.
- La configuración $r\ cv$ si $\delta(q, b) = (r, c, L)$.

Definición. Sea $M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ una MT y C, D dos configuraciones. Diremos que M *computa* D desde C si $C \vdash_M^* D$. Es decir, existen $k \in \mathbb{N}$ configuraciones $C_1, \ldots, C_k$ tales que

$$C = C_0 \vdash_M C_1 \vdash_M \cdots \vdash_M C_k = D.$$

Definición. Si una configuración está en estado q_a será *de aceptación*, si está en estado q_r será *de rechazo*. Una MT M *acepta* $w \in \Sigma^*$ si $q_0 w \vdash_M^* C$ con C de aceptación. Una MT M

rechaza $w \in \Sigma^*$ si $q_0 w \vdash_M^* C$ con C de rechazo. Si M acepta o rechaza w diremos que se *detiene* en w.

Definición. El *lenguaje reconocido* por una MT M es

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

Un lenguaje es *Turing-reconocible* si alguna máquina de Turing lo reconoce.

Definición. Un *decisor/decididor* es una MT que se detiene en toda entrada de Σ^*. Si además reconoce L, decimos que *decide* L. Un lenguaje es *Turing-decidible*, o *decidible*, si alguna MT lo decide.

Proposición. Todo lenguaje decidible es Turing-reconocible.

Definición. Sea $f\colon \Sigma_1^* \to \Sigma_2^*$ una función. Diremos que f es *computable* si existe una MT que para todo $w \in \Sigma^*$ al partir M con input w se detiene con $f(w) \in \Sigma^*$ en su cinta.

Ejercicios

1. *a)* Una MT tiene estados no terminales p, r, estado inicial p, alfabeto de entrada $\{0,1\}$ y alfabeto de cinta $\{0,1,\sqcup\}$. Sus transiciones incluyen

$$\delta(p,0) = (r,1,L), \qquad \delta(r,1) = (p,0,R), \qquad \delta(p,\sqcup) = (q_a,\sqcup,R).$$

   En los demás casos no terminales, rechaza conservando el símbolo y moviendo a la derecha. Escriba el sucesor de cada configuración $1\,p\,0$, $p\,0$ y $01\,p$. Luego escriba la computación completa sobre la entrada 0.

   b) A partir de un AFD A, construya una MT que decida $\mathcal{L}(A)$.

   c) Sea M un decisor de $L \subseteq \Sigma^*$. Construimos M' intercambiando los papeles de aceptación y rechazo. ¿Qué lenguaje decide M'? ¿Qué ocurre si M no es un decisor, sino sólo un reconocedor?

   d) ¿Toda computación arbitrariamente larga de una MT repite alguna configuración? Justifique su respuesta.

2. *a)* Diseñe un decisor para el lenguaje

$$L = \{w\#w^R : w \in \{0,1\}^*\}.$$

   ¿Es todo lenguaje libre de contexto decidible? Justifique su respuesta.

   b) Diseñe un decisor para el lenguaje

$$L = \{a^i\#a^j\#a^k : i \leq j \leq k\}.$$

   ¿Qué concluye sobre la relación entre lenguajes libres de contexto y lenguajes decidibles?

c) Una palabra no vacía $w \in \{0,1\}^*$ es *primitiva* si no existen una palabra no vacía u y un entero $k \geq 2$ tales que $w = u^k$. Por ejemplo, 01 es primitiva, mientras que 0101 no lo es.

1) Sean w, u palabras no vacías. Si $w = u^k$ para algún $k \geq 2$, ¿qué restricciones satisface $|u|$ respecto de $|w|$ y qué parte de w determina u? Use su respuesta para describir una lista finita de candidatos y diseñar un decisor del lenguaje de las palabras primitivas.
2) Explique cómo implementar su algoritmo con una sola cinta y un alfabeto finito de marcas. Indique qué información guarda en los estados y cuál conserva en la cinta. Justifique que todos los recorridos y la búsqueda completa terminan.

d) Muestre que las siguiente funciones son computables:

1) $f : \{0,1,\#\}^* \to \{0,1\}^*$ que computa la concatenación: $f(u\#v) = uv$.
2) $g : \{1\}^* \to \{1\}^*$ que computa el cuadrado en unario: $f(1^n) = 1^{n^2}$.
3) $h\colon \{1\}^* \to \{1\}^*$ que computa el logaritmo en unario: $g(1^n) = 1^{\lfloor \log_2 n \rfloor}$.

3. Sea $M = (Q, \Sigma, \Gamma, \delta, q_0, q_\text{a}, q_\text{r})$ una MT y sea $c \geq 1$ un entero fijo. Suponga que, para toda entrada w de largo n, durante toda la computación el cabezal de M permanece entre las primeras $s = c(n+1)$ celdas.

a) Encuentre una cota explícita N, en términos de $|Q|$, $|\Gamma|$ y s, para la cantidad de configuraciones distintas que pueden aparecer sobre esa entrada.

b) Demuestre que, si M se detiene en esa entrada, lo hace en menos de N pasos.

c) Construya un decisor para $\mathcal{L}(M)$ y justifique su implementación.

1. Una alarma recibe una señal via una palabra binaria $w = w_1 \cdots w_n$. El caracter w_i indica si un sensor detecta movimiento en el instante i: habrá una señal $w_i = 1$ si el sensor detecta movimiento y $w_i = 0$ en caso contrario. La alarma se activa si y sólo si hay dos instantes de la misma paridad en los que se detecta movimiento.

   a) Describa el lenguaje de las señales que activan la alarma.

   b) Diseñe un AFND, AFD y ER para el lenguaje de las señales que activan la alarma. Su AFND debe ser más pequeño que su AFD.

2. Usted juega el siguiente juego de cartas:

   - Al comienzo usted roba n cartas ordenadas con pintas en $\{\heartsuit, \spadesuit, \diamondsuit, \clubsuit\}$.
   - Usted **no** puede cambiar el orden de las cartas.
   - En cada turno, usted puede tomar dos de sus cartas consecutivas de la misma pinta y descartarlas.
   - Usted gana si logra descartar todas sus cartas.

   Por ejemplo, si robo una mano de cartas con pintas $\heartsuit\spadesuit\heartsuit\heartsuit\spadesuit\diamondsuit\diamondsuit\heartsuit$, la siguiente secuencia de movimientos gana

   $$\heartsuit\spadesuit\heartsuit\heartsuit\spadesuit\diamondsuit\diamondsuit\heartsuit \to \heartsuit\spadesuit\spadesuit\diamondsuit\diamondsuit\heartsuit \to \heartsuit\diamondsuit\diamondsuit\heartsuit \to \heartsuit\heartsuit \to \varepsilon.$$

   a) Sea L el lenguaje de las manos de cartas que ganan. Muestre que L no es regular.

   b) Muestre que L es un lenguaje libre de contexto.

3. Para un lenguaje $L \subseteq \Sigma^*$, definimos

   $$\texttt{Raiz}(L) = \{w \in \Sigma^* : ww \in L\},$$
   $$\texttt{Copia}(L) = \{ww \in \Sigma^* : w \in L\}.$$

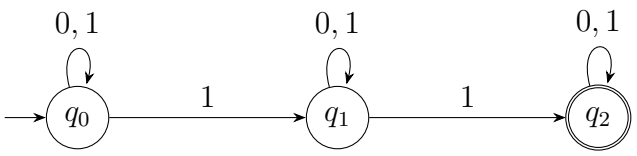
   a) Determine una expresión regular para $\texttt{Raiz}(0^*1^*)$.

   b) Muestre que si L es regular, entonces $\texttt{Raiz}(L)$ es regular.

   c) ¿Es cierto que si L es regular, entonces $\texttt{Copia}(L)$ es regular? Justifique su respuesta.

4. Un automata de aceptación única (AAU) es un AFND que acepta una palabra si y sólo si existe exactamente un camino de aceptación para esa palabra.

   a) Considere el siguiente automata:

   ¿Qué lenguaje acepta como AFND? ¿Qué lenguaje acepta como AAU?

   b) Muestre que todo lenguaje regular es aceptado por un AAU.

   c) Muestre que los lenguajes aceptados por un AAU son exactamente los lenguajes regulares.