

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. $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ 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 uaq a bvr :

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

Si M computa desde la config. C a 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 qbv :

- La configuración crv si $\delta(q, b) = (r, c, R)$.
- La configuración rcv 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 $C_1, \dots, C_k$ tales que

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

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. También se usa el nombre *recursivamente enumerable*.

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. También se usa el nombre *recursivo*.

Proposición. Todo lenguaje decidable es Turing-reconocible.

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

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

En los demás casos no terminales, rechaza conservando el símbolo y moviendo a la derecha. Escriba el sucesor de cada configuración $1p0$, $p0$ y $01p$. 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 reconocedor de $L \subseteq \Sigma^*$. Construimos M' intercambiando los papeles de aceptación y rechazo. ¿Qué lenguaje reconoce M' ? Demuestre que decide $\Sigma^* \setminus L$ si M es un decisor. Dé un contraejemplo que muestre que, para un reconocedor, M' puede no reconocer el complemento de L .
- d) ¿Toda computación arbitrariamente larga de una MT repite alguna configuración? Justifique su respuesta mediante una prueba o una máquina explícita.

2. a) Diseñe un decisor para el lenguaje

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

¿Es todo lenguaje libre de contexto decidable? 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.
3. Sea $M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ 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, incluso si la computación es infinita.
- 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. Dos escrituras que sólo difieren en blancos finales omitidos representan la misma configuración.
 - 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 mediante una cinta, marcas y recorridos finitos. La máquina M y la constante c están fijas en la construcción; la única entrada del decisor es w . El decisor puede usar más de s celdas.