

3.3. No determinismo y enumeradores

Definición. Una *MT no determinista (MTND)* es una 7-tupla $N = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$, donde

- Q es un conjunto finito de *estados*.
- Σ es el *alfabeto de entrada*, con $\sqcup \notin \Sigma$.
- Γ es el *alfabeto de cinta*, finito, con $\Sigma \cup \{\sqcup\} \subseteq \Gamma$.
- $\delta : Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L, R\}}$ es la *función de transición*.
- $q_0 \in Q$ es el *estado inicial*.
- $q_a \in Q$ es el *estado de aceptación*.
- $q_r \in Q$ es el *estado de rechazo*, con $q_r \neq q_a$.

En cada paso, elegimos una acción permitida. Sin acciones posibles: la computación se detiene sin aceptar. En q_a o q_r la computación se detiene. Cinta, inicio y borde izquierdo: como antes.

Definición. Sean N una MTND y $w \in \Sigma^*$. El *árbol de configuraciones* de N sobre w se define recursivamente: la raíz está etiquetada con la configuración inicial de N con entrada w ; cada nodo etiquetado con C tiene un hijo nuevo etiquetado con D por cada configuración D alcanzable desde C en un paso de N .

Definición. Sean N una MTND y $w \in \Sigma^*$. N *acepta* w si existe un camino en el árbol de configuraciones que llega a una configuración con estado q_a .

$$\mathcal{L}(N) = \{w \in \Sigma^* : N \text{ acepta } w\}.$$

Teorema. Sea N una MTND. Existe una MT determinista equivalente a N .

Corolario. Sea $L \subseteq \Sigma^*$. L es Turing-reconocible si y solo si alguna MTND reconoce L .

Definición. Sea N una MTND. N es un *decididor* si todas sus ramas terminan, para toda entrada.

Corolario. Sea $L \subseteq \Sigma^*$. L es decidible si y solo si algún decididor no determinista reconoce L .

Definición. Un *enumerador* sobre Σ es una MT determinista de dos cintas:

- Una *cinta de trabajo*, con las convenciones usuales.
- Una *cinta de salida*, de solo escritura: agrega símbolos de $\Sigma \cup \{\#\}$ al final, con $\# \notin \Sigma$.

Diremos que E *imprime* un string $w \in \Sigma^*$ si, en algún momento de su computación, E agrega

$w\#$ al final de la cinta de salida.

Definición. Sea E un enumerador de strings sobre Σ . El *lenguaje enumerado* por E es

$$\mathcal{L}(E) = \{w \in \Sigma^* : E \text{ imprime } w \text{ en algún momento}\}.$$

Teorema. Sea $L \subseteq \Sigma^*$. L es Turing-reconocible si y solo si algún enumerador enumera L .

Lema. Sea E un enumerador. Existe una MT que reconoce $\mathcal{L}(E)$.

Lema. Sea M un reconocedor. Existe un enumerador para $\mathcal{L}(M)$.

Ejercicios

1. a) Construya un decidor no determinista para

$$L = \{x\#y : x, y \in \{0, 1\}^*, x \neq y\}.$$

- b) Sea N una MTND tal que, para toda entrada, cada rama termina en q_a o q_r . Si intercambiamos los estados de aceptación y rechazo, ¿obtenemos una máquina que reconoce el complemento de $\mathcal{L}(N)$?
 - c) Una MTND con a lo más dos posibilidades de transición por configuración se llama *MTND binaria*. Demuestre que toda MTND puede simularse mediante una MTND binaria.
 - d) Sean $A, B \subseteq \Sigma^*$ y suponga que dispone de enumeradores para A y B . Construya un enumerador para $A \cap B$.
 - e) Dado un enumerador E , construya otro que enumere $\mathcal{L}(E)$ sin repetir ningún string.
2. Sea Σ un alfabeto y sea $\# \notin \Sigma \cup \{0, 1\}$ un separador. Un *verificador* es una MT determinista V que se detiene en toda entrada. Más aún, V rechaza toda entrada que no esté en la forma $w\#c$ con $w \in \Sigma^*$ y $c \in \{0, 1\}^*$. El lenguaje *verificado* por V es

$$L_V = \{w \in \Sigma^* : \text{existe } c \in \{0, 1\}^* \text{ tal que } V \text{ acepta } w\#c\}.$$

- a) Construya una MTND que reconozca L_V .
- b) Construya un enumerador para L_V directamente a partir de V .
- c) Demuestre que todo lenguaje Turing-reconocible es de la forma L_V para algún verificador V .
- d) Suponga ahora que existe una función computable $f : \Sigma^* \rightarrow \{0, 1\}^*$ tal que cada entrada $w \in \Sigma^*$ tiene un certificado de largo a lo más $f(w)$. Demuestre que L_V es decidible.

3. Sea Σ un alfabeto con un orden fijo entre sus símbolos. Diremos que un lenguaje $L \subseteq \Sigma^*$ está *enumerado canónicamente* si existe un enumerador que imprime los strings de L en orden creciente de tamaño y, para cada tamaño, en orden lexicográfico.
- a) Demuestre que todo lenguaje decidable puede enumerarse canónicamente.
 - b) Demuestre que todo lenguaje enumerable canónicamente es decidable.
 - c) ¿Qué dificultades surgen si se intenta enumerar canónicamente un lenguaje Turing-reconocible que no es decidable?