5.1. Tiempo y eficiencia

Definición. Sea M una MT determinista que siempre se detiene. Su *tiempo en el peor caso* es

$$T_M(n) = \max_{w \in \Sigma^n} \{\text{número de pasos de } M \text{ sobre } w\}.$$

El tamaño del input es $n = |w|$.

Definición. Sea $t : \mathbb{N} \to \mathbb{N}$. La *clase de tiempo* $\text{TIME}(t(n))$ es

$$\{L : \text{una MT determinista de una cinta decide } L \text{ en tiempo } O(t(n))\}.$$

Teorema. Sea $t(n) \geq n$. Toda MT determinista multicinta de tiempo $t(n)$ tiene una MT de una cinta equivalente de tiempo $O(t(n)^2)$.

Definición. La clase P contiene los lenguajes decidibles en tiempo polinomial por una MT determinista de una cinta.

$$\mathsf{P} = \bigcup_{k \geq 1} \text{TIME}(n^k).$$

Teorema. Todo lenguaje libre de contexto pertenece a P.

Definición. Un *verificador polinomial* para $L \subseteq \Sigma^*$ es una MT determinista V, de tiempo polinomial en $|w| + |c|$, junto con un polinomio p, tal que

$$w \in L \iff \exists c \in \{0,1\}^*, \quad |c| \leq p(|w|), \quad V(w,c) = 1.$$

c es un *certificado*.

Definición. La clase NP contiene los lenguajes que tienen un verificador polinomial.

Definición. Para una MTND que siempre termina, el *tiempo* es el máximo de pasos sobre todas las ramas y todos los inputs de largo n.

$$\text{NTIME}(t(n)) = \{L : \text{una MTND decide } L \text{ en tiempo } O(t(n))\}.$$

Teorema. Un lenguaje está en NP si y solo si alguna MTND lo decide en tiempo polinomial. En particular,

$$\mathsf{NP} = \bigcup_{k \geq 1} \text{NTIME}(n^k).$$

Proposición. $\mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{EXPTIME}$, donde

$$\mathsf{EXPTIME} = \bigcup_{k \geq 1} \mathrm{TIME}(2^{n^k}).$$

Definición. La clase coNP contiene los complementos de lenguajes de NP:

$$\mathsf{coNP} = \{L^c : L \in \mathsf{NP}\}.$$

Ejercicios

1. *a*) Describa una MT determinista de dos cintas que decida

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

   en tiempo $O(n)$, donde n es el largo total de la entrada. ¿Qué cota obtiene para una MT de una cinta mediante el teorema de simulación?

   b) Demuestre que el lenguaje

   $$\mathrm{DIVIDE} = \{\langle a, b\rangle : a, b \geq 1,\ a \text{ divide a } b\}$$

   pertenece a P, con a y b codificados en binario. Justifique el costo en función del largo de la entrada.

   c) Demuestre que P es cerrada bajo unión y complemento. Tome los complementos respecto de un mismo alfabeto Σ.

   d) Un *conjunto independiente* de un grafo simple no dirigido es un conjunto de vértices sin aristas entre ellos. Demuestre que el lenguaje de las codificaciones $\langle G, k\rangle$ para las que G tiene un conjunto independiente de tamaño al menos k pertenece a NP. Suponga $k \geq 0$ codificado en binario. Acote el largo del certificado y el tiempo de verificación.

   e) Demuestre que NP es cerrada bajo intersección.

2. Sean $a_1, \ldots, a_k \geq 1$ enteros y sea $t \geq 0$, con $k \geq 1$. En el problema de *subset-sum en unario*, la entrada es

   $$1^{a_1}\# \cdots \#1^{a_k}\#\#1^t,$$

   y se pregunta si existe $I \subseteq \{1, \ldots, k\}$ tal que $\sum_{i \in I} a_i = t$.

   a) Implemente un algoritmo de programación dinámica para decidir este problema y demuestre que pertenece a P.
   Indicación: considere los subproblemas $T[i, s]$, cuyo valor indica si existe $I \subseteq \{1, \ldots, i\}$ tal que $\sum_{j \in I} a_j = s$, para $0 \leq i \leq k$ y $0 \leq s \leq t$.

   b) Ahora suponga que todos los números de la entrada están codificados en binario. Demuestre que esta versión del problema pertenece a NP.

c) Explique por qué el algoritmo de la parte a) no permite concluir que la versión binaria de la parte b) pertenece a P y, por lo tanto, a NP.

3. Sean V un verificador determinista de tiempo polinomial en $|w| + |c|$ y $p : \mathbb{N} \to \mathbb{N}$ un polinomio.

   a) Considere el lenguaje de los pares $\langle w, u \rangle$ tales que u puede completarse hasta obtener un certificado válido para w:

   $$E = \{\langle w, u \rangle : w \in \Sigma^*,\ u \in \{0,1\}^*,\ \exists z \in \{0,1\}^*,\\ |uz| \leq p(|w|),\ V(w, uz) = 1\}.$$

   Demuestre que $E \in \mathsf{NP}$.

   b) Suponga que $\mathsf{P} = \mathsf{NP}$. Use la parte a) para construir un algoritmo de tiempo polinomial que, dado w, devuelva un certificado c con $|c| \leq p(|w|)$ y $V(w, c) = 1$, o indique que no existe tal certificado.

   c) Demuestre la recíproca: si para todo verificador V y todo polinomio p como los anteriores existe un algoritmo de búsqueda como el de la parte b), entonces $\mathsf{P} = \mathsf{NP}$.

   d) ¿Qué nos dice la equivalencia probada en esta pregunta?