

MA5201 Calculabilidad y Complejidad Computacional.

Profesor: Marcos Kiwi.

Auxiliares: Benjamín Jauregui y Bernardo Subercaseaux

Fecha: 27 de mayo de 2021.



Ingeniería Matemática
FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Auxiliar 7

Circuitos booleanos y reducciones

Definición 1. Decimos que un lenguaje $L \subseteq \{0, 1\}^*$ es **decidido por una familia de circuitos** $\mathcal{C} = (C_n)_n$ si

$$\forall n \in \mathbb{N}, \forall \omega \in \{0, 1\}^n: \omega \in L \iff C_n(\omega) = 1$$

Definición 2. Una familia de circuitos $\mathcal{C} = (C_n)_n$ se dice **uniforme** si existe una mT M que opera en espacio logarítmico tal que $M(1^n) = \langle C_n \rangle$ para todo $n \in \mathbb{N}$ ($\langle C_n \rangle$ la codificación de C_n)

Definición 3. Un lenguaje $L \subseteq \{0, 1\}^*$ tiene *complejidad tamaño-profundidad* $(f(n), g(n))$ si existe una familia de circuitos $\mathcal{C}_L = (C_n)_n$ que decide a L tal que $|C_n| = \mathcal{O}(f(n))$ y $\text{prof}(C_n) = \mathcal{O}(g(n))$

Definición 4. Para $i \geq 1$ se define la clase NC_i como los lenguajes tales que

$$NC_i = \{L: \exists k \in \mathbb{N}, L \text{ tiene tamaño profundidad } (n^k, \log^i(n)) \text{ y } \mathcal{C}_L \text{ es uniforme}\}$$

P1. Demuestre que

- a) $NC_1 \subseteq L$
- b) $NL \subseteq NC_2$

P2. Por un desperfecto técnico, el supercomputador del CMM (Leftraru) no puede ejecutar más de un trabajo (código) simultáneamente.

Los trabajos que envían las personas para ser ejecutados en Leftraru deben ser ejecutados en ciertos intervalos de tiempo predefinidos por las personas que diseñaron el código (por ejemplo, cierto trabajo debe ser ejecutado entre las 10 y 11 AM y luego entre 2 y 4 PM). Los encargados de Leftraru desean diseñar un algoritmo que les permita, mientras Leftraru tenga la falla técnica, agendar k trabajos entre los n que reciben diariamente, por lo que se contactan con ud, un entusiasta alumno de MA5201 Calculabilidad y Complejidad Computacional, para estudiar la factibilidad de este algoritmo.

Pruebe que, lamentablemente, el lenguaje

$$\text{SCHEMULTI} = \{(\langle T_1, \dots, T_n \rangle, k): \text{Se pueden aceptar al menos } k \text{ trabajos que no intersecten tpos de ejecución}\}$$

Es **NP-completo**

Obs: En la codificación de $\langle T_i \rangle$ están incluidos los tiempos en que debe ser ejecutado el trabajo T_i .

P3. Dado un grafo $G = (V, E)$, decimos que $D \subseteq V$ **domina** a los nodos de G si para todo $v \notin D$, existe un nodo vecino $\omega \in N(v)$ tal que $\omega \in D$. Decimos que dicho D es un **dominating set** (DS)

Pruebe que el lenguaje

$$\text{DOMSET} = \{\langle G, b \rangle: G \text{ tiene un DS de tamaño } \leq b\}$$

Es **NP-completo**

P4. (Propuesto) El problema de minas **PCM** es planteado como sigue: Dado un grafo $G = (V, E)$ y un conjunto de números naturales $\{p_{v_1}, \dots, p_{v_j}\}$ que son etiquetas de los vértices $\{v_1, \dots, v_j\} \subseteq V$, se debe determinar si se puede poner minas en los nodos sin etiquetas tal que todo nodo etiquetado con p_{v_i} tenga exactamente p_{v_i} vecinos con minas.

Formule el lenguaje de este problema y demuestre que es **NP-completo**

Hint: Reduzca **3SAT** a este problema.