| | |
|---|---|
| **Ma50b Complejidad Computacional** | **14 de Junio de 2008** |

# Control 3

*Prof. Cátedra: M. Kiwi* — *Prof. Auxiliar: P. Camacho*

TIEMPO: 5.5 HRS.

PROBLEMA 1:

(i).- (2.0 pts) Sea **SubISO** el lenguaje conformado por palabras de la forma $\langle H, G\rangle$ donde $H$ es subgrafo del grafo $G$.[1] Pruebe que **SubISO** es NP-completo.

(ii).- (2.0 pts) Suponga que Cook hubiese demostrado primero que **4COLOR** es NP-completo en vez de **SAT**. Usando sólo este hecho, pruebe que **SAT** es NP-duro.

(iii).- (2.0 pts) Pruebe que si P = DESPACIO$(n)$, entonces PESPACIO = P = DESPACIO$(n)$, concluya que P $\neq$ DESPACIO$(n)$.

Indicación: Use la técnica de padding vista en clase auxiliar.

PROBLEMA 2:

(i).- Pruebe que si P = NP, entonces

(i.1).- (1.2 pts) Existe un algoritmo a tiempo polinomial tal que en la entrada $\langle\varphi\rangle$, donde $\varphi$ es una fórmula Booleana en $n$ variables, determina $a_1, \ldots, a_n \in \{0,1\}$ tal que $\varphi(a_1, \ldots, a_n) = 1$ si tal asignación existe.

(i.2).- (2.4 pts) Se pueden factorizar enteros en tiempo polinomial, o sea, que existe una algoritmo a tiempo polinomial que en la entrada $n \in \mathbb{N}$ codificada en binario, $n$ compuesto, entrega como salida $m \in \mathbb{N}$ codificada en binario, $m$ un factor no trivial de $n$.

Indicación: Utilice búsqueda binaria.

(ii).- (2.4 pts) Pruebe que $2SAT$ es NL-completo.

Indicación: Recuerde que $x \implies y$ ssi $\neg x \vee y$.

PROBLEMA 3: Una máquina de Turing alternante (mTA) es una máquina de Turing no-determinista cuyos estados se dividen en estados universales y existenciales. Cada nodo

[1] Se dice que $S = (U, F)$ es subgrafo de $G = (V, E)$ si $U \subseteq V$ y $vw \in F$ para todo $vw \in E$ tal que $v, w \in U$.

del árbol de cómputos de una mTA representa una configuración de la máquina en algún estado que puede ser del tipo universal o existencial (etiquetado por $\forall$ o $\exists$ respectivamente). Decimos que un nodo etiquetado por $\forall$ (respectivamente $\exists$) acepta si todos sus hijos (respectivamente alguno de sus hijos) aceptan. Si la raíz del árbol de cómputos de una mTA en la entrada $\omega$ es un nodo de aceptación, decimos que la mTA acepta.

Decimos que una mTA tiene $k$ niveles de alternancia comenzando con $\exists$ (respectivamente $\forall$), denotada $k$-mTA$_\exists$ (respectivamente $k$-mTA$_\forall$) si cualquiera que sea su entrada, en todas sus ramas de cálculo las etiquetas en dicha rama son $Q_1^* Q_2^* Q_3^* \dots Q_k^*$ donde $Q_i \in \{\forall, \exists\}$, $Q_{i+1} \neq Q_i$ y $Q_1 = \exists$ (respectivamente $Q_1 = \forall$).

Las nociones de complejidad de espacio y tiempo asociados a una mTA son las mismas que las de cualquier máquina de Turing nodeterminista. Se definen las clases

$$\begin{aligned}\Sigma_k^P &= \bigcup_{c\in\mathbb{N}} \{L : L \text{ es decidido por una } k\text{-mTA}_\exists \text{ a tiempo } O(n^c)\}\ ,\\ \Pi_k^P &= \bigcup_{c\in\mathbb{N}} \{L : L \text{ es decidido por una } k\text{-mTA}_\forall \text{ a tiempo } O(n^c)\}\ .\end{aligned}$$

(i).- (2.0 pts) Sea $TQBF_k$ el lenguaje tal que $\langle\phi\rangle \in TQBF_k$ si y sólo si $\phi$ es una fórmula Booleana totalmente cuantificada verdadera tal que $\phi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n C(x_1, \dots, x_n)$ donde $Q_1 = \exists$, $x_1, \dots, x_n$ son variables Booleanas, $C$ es un circuito Booleano, y hay a lo más $k-1$ alternancias de cuantificadores en la secuencia $Q_1 Q_2 \dots Q_n$.

Pruebe que $TQBF_k$ es completo para $\Sigma_k^P$, i.e. $TQBF_k \in \Sigma_k^P$ y que para todo $L \in \Sigma_k^P$ se tiene que $L \leq_P TQBF_k$.

(ii).- Se define la jerarquía polinomial como $\mathrm{PH} = \bigcup_{k\in\mathbb{N}} \Sigma_k^P$.

(ii.1).- (0.5 pts) Pruebe que $\mathrm{PH} = \bigcup_{k\in\mathbb{N}} \Pi_k^P$.

(ii.2).- (1.5 pts) Pruebe que si $\Sigma_k^P = \Sigma_{k+1}^P$, entonces la jerarquía polinomial colapsa a su $k$-ésimo nivel, i.e. $\mathrm{PH} = \Sigma_k^P$.

(iii).- Pruebe que:

(i.1).- (0.6 pts) $\Sigma_1^P = \mathrm{NP}$.

(i.2).- (0.6 pts) $\Sigma_k^P \subseteq \Pi_{k+1}^P$ y que $\Pi_k^P \subseteq \Sigma_{k+1}^P$.

(i.3).- (0.8 pts) $\Pi_k^P = co\Sigma_k^P$.