Ma50b Complejidad Computacional **14 de Mayo de 2009**

Control 2

Prof. Cátedra: M. Kiwi *Prof. Auxiliar: I. Fantini, T. González*

TIEMPO: 5.0 HRS.

PROBLEMA 1:

(i).- (3.0 pts) Sea QuadraticEq el lenguaje de los $\langle A^{(1)}, A^{(2)}, \ldots, A^{(m)}, b\rangle$ donde $A^{(k)} \in \mathbb{F}_2^{n\times n}$, $b \in \mathbb{F}_2^m$, y tales que el siguiente sistema de ecuaciones cuadráticas tiene solución en $\mathbb{F}_2^n$,

$$\sum_{i,j=1}^{n} A^{(k)}_{i,j} x_i x_j = b_k\,, \quad k = 1,\ldots,m.$$

Pruebe que QuadraticEq es NP-completo.

(ii).- (3.0 pts) Sea FeedbackArc el lenguaje de los $\langle G,k\rangle$ donde $G = (V,E)$ es un digrafo, $k \in \mathbb{N}$, y tal que existe un $F \subseteq E$, $|F| \leq k$, para el cual $G \setminus F = (V, E \setminus F)$ es un digrafo acíclico. Pruebe que FeedbackArc es NP-completo.

Indicación: Reduzca desde VertexCover.

PROBLEMA 2: Sea $L \subseteq \{0,1\}^*$. Se dice que $L \in \mathrm{P/poli}$ si existe una máquina de Turing determinista M a tiempo polinomial, un polinomio p y una secuencia de consejos $(\alpha_n : n \in \mathbb{N})$, $\alpha_n \in \{0,1\}^*$ donde $|\alpha_n| \leq p(n)$, tal que para todo $\omega \in \{0,1\}^n$, se tiene que $\omega \in L$ si y solo si M acepta $\langle \omega, \alpha_n\rangle$.

Todos los circuitos Booleanos a los que se hace referencia en lo que sigue son con puertas lógicas solo del tipo $\vee$, $\wedge$ y $\neg$.

Se dice que L es decidido por una familia de circuitos Booleanos $(C_m : m \geq 1)$ de tamaño polinomial si existe un polinomio q tal que $|C_m| \leq q(m)$ y para todo $\omega \in \{0,1\}^m$, se tiene que $\omega \in L$ si y solo si $C_m(\omega) = 1$.

(i).- (2.5 pts) Pruebe que si $L \in \mathrm{P}$, entonces existe una familia de circuitos Booleanos de tamaño polinomial que decide L.

(ii).- (2.5 pts) Pruebe que $L \in \mathrm{P/poli}$ si y solo si existe una familia de circuitos Booleanos de tamaño polinomial que decide L. Concluya que $\mathrm{P} \subseteq \mathrm{P/poli}$.

(iii).- (1.0 pts) Lo establecido en (ii) sugiere que una estrategia para probar que $\mathrm{P} \neq \mathrm{NP}$ es mostar que existe un lenguaje en NP que no puede ser decidido por una familia de circuitos de tamaño polinomial. ¿Qué lenguajes sería natural elegir para intentar la estrategia descrita? Argumente informalmente.

PROBLEMA 3:

(i).- (1.5 pts) Se dice que un lenguaje L es autoreducible hacia abajo si existe una máquina de Turing M que en entradas ω de largo n puede decidir si ω pertenece a L en base a consultas sobre la membresia en L de secuencias de largo a lo más $n-1$. Pruebe que SAT, para la codificación natural de las fórmulas Booleanas, es autoreducible hacia abajo.

(ii).- (2.0 pts) Sean (T,p,c) tales que $T=(V,E)$ es un árbol binario enraizado y totalmente balanceado de profundidad a lo más n (por lo tanto, con potencialmente 2^n-1 nodos), p un polinomio, y c una función que le asocia a cada nodo interno de T un color en $\{0,\ldots,p(n)\}$ y a cada hoja de T un valor en $\{acept, rech\}$. Sea S el conjunto de hojas de T donde c evalua a $acept$ (notar que S es potencialmente vacío). Sea A el conjunto de antecesores en T de las hojas en S.[1] Asumiendo que $c(A)\cap c(V\setminus A)=\emptyset$, muestre como encontrar en tiempo polinomial en n una hoja h de T tal que $c(h)=acept$ o establecer que tal hoja no existe. Suponga que en tiempo polinomial en n, dado un nodo v de T se puede determinar el valor de c en v, y el hijo izquierdo, el hijo derecho, o el padre de v.

Indicación: Considere el conjunto Izq de nodos internos de T a la "izquierda" del conjunto Cam de nodos internos de T en un camino desde la raíz de T a la hoja de más a la "izquierda" de T donde c toma el valor $acept$. Pruebe que si hay $n+1$ nodos en $Izq\cup Cam$ con el mismo color, entonces dicho color no puede ser uno de los asignados a los nodos en A. Use esto para hacer eficientemente una búsqueda en T.

(iii).- (2.5 pts) Un lenguaje U se dice unario si $U\subseteq 1^*$. Pruebe que si existe U lenguje unario y NP-completo, entonces P = NP.

Indicación: Observe que si existe una reducción de SAT a U a tiempo $p(n)$, donde p es un polinomio, entonces la reducción lleva instancias de SAT de tamaño n a una instancia 1^i de U donde $0\leq i\leq p(n)$.

[1] Considere que un nodo ES antecesor de si mismo.