Ma50b Complejidad Computacional **14 de Mayo de 2005**

Pauta Control 2

Prof. Cátedra: M. Kiwi *Prof. Auxiliar: J. Soto*

PROBLEMA 1:

(i.1).- Las máquinas de Turing con acceso a un oráculo dado tienen una descripción finita, al igual que en el caso de máquinas de Turing estándar. Se concluye que hay una cantidad numerable de máquinas de Turing con oráculo A_{mT}. Por otra parte sabemos que hay una cantidad no–numerable de lenguajes. Luego, existen lenguajes que no son reconocibles por máquinas de Turing con oráculo A_{mT}.

(i.2).- La idea aquí es argumentar exactamente como en el caso de la demostración de indecibilidad de A_{mT}. Específicamente, asumiremos que Z es decidible por una máquina de Turing H con oráculo A_{mT} y obtendremos una contradicción. Construiremos una máquina de Turing S con oráculo A_{mT} que usa H como subrutina. Esta máquina recibe una entrada de la forma $\langle M\rangle$ con M máquina de Turing con oráculo A_{mT} y determina lo que hace $M^{A_{mT}}$ en la entrada $\langle M\rangle$. Una vez determinado lo que M hace, la máquina S hace lo contrario, i.e. rechaza si $M^{A_{mT}}$ acepta y no acepta si $M^{A_{mT}}$ acepta. Formalmente,

$S^{A_{mT}}$ = "En la entrada σ,

Verifica que $\sigma = \langle M\rangle$ con M máquina de Turing con oráculo.

Simula H con oráculo A_{mT} en la entrada $\langle M, \langle M\rangle\rangle$ (para la simulación S usa su oráculo)

– Acepta si $H^{A_{mT}}$ no acepta, y rechaza si $H^{A_{mT}}$ acepta."

Resumiendo,

$$S^{A_{mT}}(\langle M\rangle) = \begin{cases} acep., & \text{si } M^{A_{mT}} \text{ no acepta } \langle M\rangle, \\ rech., & \text{si } M^{A_{mT}} \text{ acepta } \langle M\rangle. \end{cases}$$

Se obtiene la siguiente contradicción,

$$S^{A_{mT}}(\langle S\rangle) = \begin{cases} acep., & \text{si } S^{A_{mT}} \text{ no acepta } \langle S\rangle, \\ rech., & \text{si } S^{A_{mT}} \text{ acepta } \langle S\rangle. \end{cases}$$

Luego, H no existe y por lo tanto Z no es decidible por una máquina de Turing con oráculo A_{mT}.

(ii).- Sea M una máquina de Turing que decide L en tiempo $O(n^k)$ para algún $k \in \mathbb{N}$. Construiremos una máquina de Turing M' que en la entrada $\omega = \omega_1\omega_2\ldots\omega_n$ construye en n etapas una tabla de entradas (i, j), $1 \le i \le j \le t$. El objetivo es que al finalizar la construcción, la tabla contenga en la entrada (i, j) el valor 1 si y sólo si la palabra $\omega_i \ldots \omega_j \in L^*$. Específicamente, en la t-ésima etapa, M' calcula todas las entradas (i, j) de la tabla para las que $j - i = t$. Inicialmente $t = 0$ y M' coloca la entrada (i,i) de la tabla en 0 o 1 dependiendo de si $M(\omega_i) = \textit{rechaza}$ o $M(\omega_i) = \textit{acepta}$, i.e., si $\omega_i \notin L$ o $\omega_i \in L$ respectivamente. En la etapa $t+1$, para cada $i \le j$, M' verifica si para algún $s \in \{i, \ldots, j\}$ las entradas (i,s) y $(s+1, j)$ son ambas iguales a 1. Si esto ocurre para algún s o si $M(\omega_i, \ldots \omega_j) = \textit{acepta}$ entonces M' fija la entrada (i, j) en 1.

Concluidas las n etapas, M' acepta si y sólo si la entrada $(1,n)$ de la tabla contiene un 1. Inductivamente se prueba que terminada la t–ésima etapa, la entrada (i,j) de la tabla es igual a 1 si y sólo si $\omega_i \ldots \omega_j \in L^t$ para algún $m \leq n$. Dado que $\omega \in L^*$ si y sólo si $\omega \in L^m$ para algún $m \leq n$, se concluye que M' acepta ω si y sólo si $\omega \in L^*$.

Finalmente, observar que dado que cada etapa toma tiempo $O(n^2 \cdot n^k)$ y como hay $O(n)$ etapas, sigue que M' es a tiempo polinomial.

PROBLEMA 2:

(i).- Sea $L \in \text{RTIEMPO}(T(n))$ y sea M la mT probabilista a tiempo $T(n)$ tal que

$$\begin{aligned} \omega \in L \quad &\Longrightarrow \quad \mathbb{P}_\rho\,(M \text{ acepta } \omega) \geq 1/2\,, \\ \omega \notin L \quad &\Longrightarrow \quad \mathbb{P}_\rho\,(M \text{ rechaza } \omega) = 1\,. \end{aligned}$$

Sea N la mT no–determinista que simula M pero cada vez que M accede a una nueva celda de su cinta aleatoria la máquina N en forma no–determinista adivina el contenido de la celda (0 o 1) y utiliza este valor como el bit aleatorio que M hubiese obtenido. La máquina N acepta si M entra en un estado de aceptación y rechaza si M entra en un estado de rechazo. Sigue que N es a tiempo $O(T(n))$ y que

$$\begin{aligned} \omega \in L \quad &\Longrightarrow \quad \mathbb{P}_\rho\,(M \text{ acepta } \omega) \geq 1/2\,, \\ &\Longrightarrow \quad \exists \rho \in \{0,1\}^{T(|\omega|)} \text{ contenido de la cinta aleatoria de } M \text{ que causa} \\ &\qquad\quad \text{que } M \text{ acepte } \omega \\ &\Longrightarrow \quad \text{existe una secuencia de transiciones no–deterministas de } N \text{ que} \\ &\qquad\quad \text{la llevan a aceptar } \omega \\ &\Longrightarrow \quad N \text{ acepta } \omega. \end{aligned}$$

Análogamente se verifica que

$$\begin{aligned} \omega \notin L \quad &\Longrightarrow \quad \mathbb{P}_\rho\,(M \text{ rechaza } \omega) = 1\,, \\ &\Longrightarrow \quad \forall \rho \in \{0,1\}^{T(|\omega|)} \text{ contenido de la cinta aleatoria de } M \text{ causa} \\ &\qquad\quad \text{que } M \text{ rechace } \omega \\ &\Longrightarrow \quad \text{toda secuencia de transiciones no–deterministas de } N \text{ la} \\ &\qquad\quad \text{llevan a rechazar } \omega \\ &\Longrightarrow \quad N \text{ rechaza } \omega. \end{aligned}$$

Luego, tenemos que N decide L en tiempo $O(T(n))$, i.e., $L \in \text{NDTIEMPO}(T(n))$.

(ii).- Sean

$$\begin{aligned} MAX-SAT_{\leq} \quad &= \quad \left\{ \langle \varphi, k \rangle : \begin{array}{l} \varphi \text{ es una formula Booleana en forma conjuntiva normal cuyo} \\ \text{número máximo de cláusulas que se puede satisfacer es a lo más } k \end{array} \right\}, \\ MAX-SAT_{\geq} \quad &= \quad \left\{ \langle \varphi, k \rangle : \begin{array}{l} \varphi \text{ es una formula Booleana en forma conjuntiva normal cuyo} \\ \text{número máximo de cláusulas que se puede satisfacer es al menos } k \end{array} \right\}. \end{aligned}$$

Vía una reducción desde $3SAT$, es fácil establecer que $MAX-SAT_{\geq}$ es NP–completo. Análogamente, se tiene que $MAX-SAT_{\leq}$ es coNP–completo.

Observar que NP es cerrado bajo intersección. Luego, si $\mathrm{NP} = co\mathrm{NP}$, entonces $MAX-SAT_{\leq} \in \mathrm{NP}$ y por lo tanto

$$MAX-SAT = MAX-SAT_{\leq} \cap MAX-SAT_{\geq} \in \mathrm{NP}.$$

Supongamos ahora que $MAX-SAT \in \mathrm{NP}$. Afirmamos que $MAX-SAT_{\leq} \in \mathrm{NP}$, lo que implica que $co\mathrm{NP} \subseteq \mathrm{NP}$. En efecto, si $MAX-SAT \in \mathrm{NP}$, entonces existe un verificador a tiempo polinomial para $MAX-SAT$ tal que $\langle \varphi, k \rangle \in MAX-SAT$ si y sólo si existe un certificado π tal que $V(\langle \varphi, k \rangle, \pi) = acepta$. Sea entonces el verificador V' tal que $V'(\langle \varphi, k \rangle, \pi_k, \ldots, \pi_m) = acepta$ si y sólo si $V(\langle \varphi, i \rangle, \pi_i) = acepta$ para algún $i = 1, \ldots, k$. Es fácil ver que V' es un verificador a tiempo polinomial para $MAX-SAT_{\leq}$, es decir $MAX-SAT_{\leq} \in \mathrm{NP}$. Análogamente se puede establecer que $MAX-SAT_{\geq} \in co\mathrm{NP}$ y por lo tanto $\mathrm{NP} \subseteq co\mathrm{NP}$. En resumen, si $MAX-SAT \in \mathrm{NP}$, entonces $co\mathrm{NP} = \mathrm{NP}$.

PROBLEMA 3:

(i).- Basta definir la reducción f que en $\langle \varphi \rangle$ con φ fórmula Booleana en forma conjuntiva normal, toma el valor $\langle G, nm^2 + 2m \rangle$ donde G es el grafo construido como se sugiere en la indicación, n es el número de variables de φ y m es el número de cláusulas de φ. Claramente la reducción es a tiempo polinomial. Falta ver que

$$\langle \varphi \rangle \in NAESAT \Longleftrightarrow \langle G, nm^2 + 2m \rangle \in MAX-CUT\,.$$

En efecto, $\langle \varphi \rangle \in NAESAT$, entonces al definir $S \subseteq V(G)$ como el conjunto de nodos etiquetados por literales verdaderos, se verifica que $|\delta(S)| = nm^2 + 2m$ (en efecto, cada *gadget* asociado a una variable aporta m^2 arcos al corte y cada cláusula otros 2 arcos).

Por otro lado, observar que un *gadget* asociado a una variable puede aportar a lo más m^2 arcos al corte. Además, un *gadget* asociado a una cláusula (un triángulo), puede aportar a lo más 2 arcos al corte. Luego, un grafo G como el obtenido al realizar la reducción tiene un corte máximo de tamaño $nm^2 + 2m$. Si un corte alcanza este tamaño, necesariamente:

1. Debe dejar todos los nodos etiquetados por una variable x en un "lado" del corte distinto a aquel donde quedan los nodos etiquetados por $\bar{x}$.
2. Debe dejar al menos un vértice de cada *gadget* asociado a un cláusula en cada "lado" del corte.

Luego, si $\langle G, nm^2 + 2m \rangle \in MAX-CUT$, entonces existe $S \subseteq V(G)$ tal que $|\delta(S)| \geq nm^2 + 2m$. Si le asignamos el valor verdadero a todos los literales que quedan a un "lado" del corte y falso a los que quedan del otro "lado", obtendremos una asignación consistente (por la condición 1). Esta asignación además deja un literal verdadero y un literal falso por cada cláusula de φ (por la condición 2).

Falta ver que $MAX-CUT$ está en NP. Para ello basta observar que un certificado de pertenencia de $\langle G, k \rangle$ en $MAX-CUT$ sería un conjunto $S \subseteq V(G)$ y que la verificación de dicho certificado puede hacerse eficientemente, ya que sólo involucra verificar que el número de arcos en $E(G)$ que tiene un único extremo en S es al menos k.

(ii).- Podemos asociar a cada estudiante el subconjunto de los exámenes que debe tomar. El problema planteado corresponde entonces al problema de decisión asociado al lenguaje

$$SCHED = \left\{ \langle \{S_i\}_{i \in I}, h \rangle : \begin{array}{l} \exists C_1, \ldots, C_h \text{ partición de } \Omega = \cup_{i \in I} S_i \text{ tal que para todo } S_i \text{ inter-} \\ \text{secta a lo más a un } C_j \end{array} \right\}.$$

Consideremos la aplicación $f(\langle G\rangle) = \langle \{S_e\}_{e\in E}, 3\rangle$ donde G es un grafo y para $uv \in E(G)$ se define $S_{uv} = \{u, v\}$. En otras palabras, vemos cada nodo y cada arco de G como un examen y como un estudiante respectivamente, e implícitamente fijamos que cada estudiante toma dos exámenes (aquellos correspondientes a los extremos del arco que lo representa). Claramente, f se puede calcular en tiempo polinomial.

Falta ver que f efectivamente reduce $3COLOR$ a $SCHED$. A un 3 coloreamiento χ de G le corresponde de manera natural una partición $\{C_1, C_2, C_3\}$ de los exámenes. Cada conjunto de exámenes de esta partición puede ser programada en un día distinto puesto que si algún estudiante tuviese que tomar dos exámenes el mismo día entonces χ no sería un coloreamiento. Por otro lado, supongamos que $\langle \{S_i\}_{i\in I}, 3\rangle$ está en $SCHED$. Existe entonces una partición $\{C_1, C_2, C_3\}$ de $V(G) = \cup_{i\in I} S_i$ tal que todo S_i intersecta a lo más a un C_j. En otras palabras, $\chi : V(G) \to \{1, 2, 3\}$ tal que $\chi(v) = j$ si y sólo si $v \in C_j$ es un 3–coloreamiento de G.