MA50B Complejidad Computacional	**22 de Junio, 2002**

Pauta Control 3

Profesor: M. Kiwi	**Auxiliar: E. Moreno**

PROBLEMA 1:

(i).- La misma razón por la cual se tiene que $3SAT \in \text{NP}$ conlleva que $B3SAT \in \text{NP}$.

Para ver que $B3SAT$ es NP–duro basta probar que $3SAT \leq_P B3SAT$. En efecto, sea $\phi = \phi(x_1, \ldots, x_n)$ fórmula Booleana en forma 3 conjuntiva normal. Supongamos que la variable x_i aparece k_i veces en ϕ. Sustituyendo la j–ésima aparición de x_i por la variable Booleana $x_{i,j}$, agregándole a ϕ las cláusulas

$$\begin{aligned}\bigwedge_{j=1}^{k_i-1} (\neg x_{i,j} \vee x_{i,j+1}) \wedge (\neg x_{i,k_i} \vee x_{i,1}) &\equiv \bigwedge_{j=1}^{k_i-1} (x_{i,j} \Rightarrow x_{i,j+1}) \wedge (x_{i,k_i} \Rightarrow x_{i,1}) \\ &\equiv x_{i,1} = x_{i,2} = \ldots = x_{i,k_i},\end{aligned}$$

y repitiendo el procedimiento para todo $i \in \{1, \ldots, n\}$, se obtiene una fórmula ϕ'. Notar que ϕ' queda en forma 3 conjuntiva normal en que cada variable. Más aún, cada literal aparece a lo más 3 y 2 veces respectivamente. Además,

$$\langle \phi \rangle \in 3SAT \Longleftrightarrow \langle \phi' \rangle \in B3SAT.$$

Como la transformación de ϕ a ϕ' es claramente a tiempo polinomial, sigue que $3SAT \leq_P B3SAT$.

(ii.1).- Veamos primero que a lo más 7 de las 10 cláusulas mencionadas pueden satisfacerse simultáneamente. Si $(\neg x \vee \neg y)$, $(\neg y \vee \neg z)$ y $(\neg x \vee \neg z)$ son falsas, estamos listos. Sin pérdida de generalidad, por la simetría existente entre las variables $\{x, y, z\}$, asumamos que sólo la primera de éstas cláusulas es falsa, i.e., que $x = y = 1$ y $z = 0$. Luego, las dos cláusulas (z) y $(\neg x \vee \neg y)$ son falsas. Si $w = 0$ tendríamos una tercera cláusula falsa. Supongamos entonces que $w = 1$. Sigue que $(z \vee \neg w) = 0$ y establece el resultado deseado.

Si $(x \vee y \vee z) = 1$, entonces ocurre uno de los siguientes casos:

- Una de las variables $\{x, y, z\}$ es verdadera, digamos $x = 1$ y $y = z = 0$. Tomando $w = 0$ obtenemos 7 cláusulas que evalúan a 1 (una en la primera, tres en la segunda y tres en la tercera fila de la tabla de cláusulas).

- Dos de las variables $\{x, y, z\}$ son verdaderas, digamos $x = y = 1$ y $z = 0$. Tomando $w = 0$ obtenemos 7 cláusulas que evalúan a 1 (dos en la primera, dos en la segunda y tres en la tercera fila de la tabla de cláusulas).
- Todas las variables $\{x, y, z\}$ son verdaderas. Tomando $w = 1$ obtenemos 7 cláusulas que evalúan a 1 (cuatro en la primera, ninguna en la segunda y tres en la tercera fila de la tabla de cláusulas).

Luego, una asignación de valores de verdad que satisface $(x \vee y \vee z)$ puede ser extendida para satisfacer 7 y no más de las mencionadas cláusulas.

(ii.2).- Observar que si $(x \vee y \vee z) = 0$, entonces de las 10 cláusulas de la parte (ii.1) 4 o 6 son verdaderas dependiendo de si $w = 1$ o $w = 0$. En cualquier caso, a lo más 6 de ellas evalúan a verdadero.

Para ver que $MAX2SAT$ es NP–duro, mostraremos que $3SAT \leq_P MAX2SAT$. En efecto, sea ϕ una fórmula Booleana en forma 3 conjuntiva normal con m cláusulas. Construimos una fórmula ϕ' reemplazando la i–ésima cláusula de ϕ, digamos $(a \vee b \vee c)$ donde $\{a, b, c\}$ son tres literales cualesquiera, por las 10 cláusulas de la parte (ii.1) haciendo $x = a$, $y = b$, $z = c$, y $w = w_i$ donde w_i es una variable Booleana adicional que introducimos. Sigue que ϕ' es una fórmula Booleana en forma 2 conjuntiva normal con $10m$ cláusulas. Por (ii.1), sigue que

$$\langle \phi \rangle \in 3SAT \Longleftrightarrow \langle \phi', 7m \rangle \in MAX2SAT.$$

Como la transformación de $\langle \phi \rangle$ a $\langle \phi' \rangle$ es fácilmente realizable en tiempo polinomial, concluimos que $MAX2SAT$ es NP–duro.

Por otro lado, un certificado para la pertenencia de $\langle \phi', k \rangle \in MAX2SAT$ es una asignación de valores de verdad $a_1, \ldots, a_n$ a las variables de ϕ. El proceso de verificación de que $a_1, \ldots, a_n$ satisface al menos k cláusulas de ϕ es a tiempo polinomial. En efecto, evaluar fórmulas Booleanas y contar es un proceso que pueden hacer las máquinas de Turing en forma eficiente. En resumen, $MAX2SAT \in$ NP.

(iii.1).- Sea $\langle \phi \rangle$ la codificación de una fórmula Booleana en forma conjuntiva normal. Sean $x_1, \ldots, x_n$ las variables de ϕ. Consideremos el algoritmo $\mathcal{A}$ que en la entrada $\langle \phi \rangle$ elige al azar valores $a_1, \ldots, a_n \in \{0, 1\}$. Sea una cláusula C cualquiera de ϕ, esta debe contener algún literal, digamos x. Sea i tal que $x = x_i$ o $x = \neg x_i$. La probabilidad que a_i haga que $x = 1$ es claramente $1/2$. Luego, la probabilidad que C evalúe a 1 en $a_1, \ldots, a_n$ es al menos $1/2$.

Si $X_{\phi,j}$ es la variable aleatoria indicatriz del evento "la j-ésima cláusula de ϕ evalúa a 1 en $a_1, \ldots, a_n$", sigue que $\mathbb{E}\left(X_{\phi,j}\right) \geq 1/2$. Por lo tanto, si $X_\phi = \sum_{j=1}^{m} X_{\phi,j}$ donde m es el número de cláusulas de ϕ, se tiene que $\mathbb{E}\left(X_\phi\right) \geq m/2$ es el valor esperado del número de cláusulas para el cual $\mathcal{A}$ encuentra una asignación que las hace cierta. Claramente, $\mathcal{A}$ es un algoritmo probabilista a tiempo polinomial, de hecho es a tiempo lineal.

(iii.2).- Sin pérdida de generalidad podemos suponer que en ϕ no hay cláusulas con literales repetidos, i.e., del tipo $(x \vee x)$, ni cláusulas done aparece una variable y su negación, i.e., del tipo $(x \vee \overline{x})$. De lo contrario, las del primer tipo pueden reemplazarse por x y las del segundo tipo eliminarse, pues no importa cual sea la asignación de valores de verdad siempre quedarán satisfechas.

Usando la notación introducida en la parte (iii.1), vemos que para los ϕ con las características recién mencionadas, $\mathbb{E}\left(X_\phi\right)$ se puede calcular eficientemente puesto que

$$\mathbb{E}\left(X_\phi\right) = \frac{1}{2}N_1(\phi) + \frac{3}{4}N_2(\phi)\,,$$

donde $N_i(\phi)$ es el número de cláusulas de ϕ con i literales.

De manera similar se pueden procesar tanto $\phi|_{x_1=0}$ como $\phi|_{x_1=1}$ y determinar $\mathbb{E}\left(X_{\phi|_{x_1=0}}\right)$ y $\mathbb{E}\left(X_{\phi|_{x_1=1}}\right)$. Haciendo $a_1 = b$ donde b maximiza $\mathbb{E}\left(X_{\phi|_{x_1=b}}\right)$ y repitiendo el proceso recursivamente se obtiene finalmente una asignación de valores de verdad $a_1, \ldots, a_n$ que satisface al menos $\mathbb{E}\left(X_\phi\right)$ cláusulas. Luego, por (iii.1), al menos la mitad de las cláusulas de ϕ.

(iv.1).- Sea M la máquina de Turing no–determinista que en la entrada $\langle \phi \rangle$, ϕ fórmula Booleana en las variables $x_1, \ldots, x_n$ adivina usando su no–determinismo $a_1, \ldots, a_n \in \{0,1\}$ (dando lugar a 2^n ramas de cálculo) y acepta si $\phi(a_1, \ldots, a_n) = 1$. Claramente, $\langle \phi \rangle \in \oplus SAT$ si y sólo si M en $\langle \phi \rangle$ tiene un número par de ramas de cálculo que llevan a aceptar.

(iv.2).- Sea ϕ una fórmula Booleana en las variables $x_1, \ldots, x_n$. Sea ϕ' otra fórmula obtenida agregándole a cada cláusula de ϕ la variable z, i.e., si C es una cláusula de ϕ entonces ϕ' tendrá una cláusula $C' = C \vee z$. También agregamos, para todo $i \in \{1, \ldots, n\}$, las cláusulas $(z \implies x_i) \equiv (\neg z \vee x_i)$. Observar que $\phi(a_1, \ldots, a_n) = 1$ si y sólo si $\phi'(a_1, \ldots, a_n, 0) = 1$, y que la única asignación de valores de verdad para la cual ϕ' es verdadera y $z = 1$ es $(x_1, \ldots, x_n, z) = (1, \ldots, 1)$. Luego, ϕ' se puede satisfacer en exactamente una asignación de valores de verdad más que el número de asignaciones que satisfacen ϕ.

Notar que la construcción descrita puede realizarse en tiempo polinomial y que $\langle \phi' \rangle \in \oplus SAT$ si y sólo si $\langle \phi \rangle\ \overline{\not\oplus SAT}$.

Sea M una máquina de Turing no–determinista asociada a $\oplus SAT$ con las características que se derivan de la pertenencia de $\oplus SAT$ en $\oplus$P. La máquina de Turing no–determinista que en la entrada $\langle \phi \rangle$ obtiene $\langle \phi' \rangle$ y luego simula M coloca a $\overline{\oplus SAT}$ en $\oplus$P.