Ma5201 Complejidad Computacional **02 de Junio de 2012**

Pauta Control 2

Prof. Cátedra: M. Kiwi *Prof. Auxiliar: P. Muñoz, D. Salas*

PROBLEMA 1:

(i).- Sea $\langle G,k\rangle$ una instancia de IndepSet, es decir tal que $G=(V,E)$ es un grafo y $k\in\mathbb{N}$. A la referida instancia le asociaremos una instancia $f(\langle G,k\rangle)$ de SubComb como se describe a continuación. Los objetos a subastar serán los arcos E. Identificamos a cada nodo $v\in V$ con un oferente. La lista de objectos deseados por el oferente $v\in V$ será el conjunto $S_v=\{e\in E\,:\,e \text{ incide en } v \text{ en } G\}$. El monto que el oferetne v está dispuesto a pagar por quedarse con el lote S_v será $\omega_v=1$. La instancia $f(\langle G,k\rangle)$ será $\langle((S_v,\omega_v):v\in V),k\rangle$. Dado $\langle G,k\rangle$ generar $f(\langle G,k\rangle)$ corresponde esencialmente a determinar los arcos incidentes en cada nodo de G, lo que se puede implementar eficientemente en el modelo RAM en $O(|V(G)|+|E(G)|)$ pasos (asumiendo que el grafo G se codifica como una lista de adyacencia), luego f es calculable en tiempo polinomial por una máquina de Turing.

Veamos ahora que $\langle G,k\rangle\in$ IndepSet si y solo si $f(\langle G,k\rangle)\in$ SubComb. En efecto, supongamos que $I\subseteq V(G)$ es un conjunto independiente en G de tamaño al menos k. Consideremos los conjuntos S_v con $v\in I$. Se tiene que si $u\neq v$, entonces S_u y S_v son disjuntos, pues de lo contrario existiria un arco $e\in E(G)$ incidente en $u,v\in I$, contradiciendo que I es conjunto independiente. Como $\sum_{v\in I}\omega_v=|I|\geq k$, sigue que $f(\langle G,k\rangle)\in$ SubComb. Supongamos ahora que $f(\langle G,k\rangle)\in$ SubComb. Se tiene entonces que existe $I\subseteq V$ tal que $(S_v:v\in I)$ es una secuencia disjunta y además $|I|=\sum_{v\in I}\omega_v\geq k$. Sigue que I es de tamaño al menos k y que además es conjunto independiente en G, puesto que de lo contraio existirían $u\neq v$ tales que $u,v\in I$ y $uv\in E(G)$, implicando que $uv\in S_u\cap S_v\neq\emptyset$, contradiciendo el hecho que los S_v con $v\in I$ son disjuntos. En resumen, hemos establecido que f es una reducción a tiempo polinomial de IndepSet a SubComb.

Solo falta comprobar que SubComb está en NP. Para ello basta considerar el certificado $S_{i_1},\ldots,S_{i_\ell}$ y verificar que los S_{i_j}'s son disjuntos y que $\sum_{j=1}^{\ell}\omega_{i_j}\geq k$.

(ii).- Sea $\langle G,k\rangle$ una instancia de VC, es decir tal que $G=(V,E)$ es un grafo y $k\in\mathbb{N}$. A la referida instancia le asociaremos una instancia $f(\langle G,k\rangle)$ de MaxMCut como se describe a continuación. Denotaremos por $\text{grado}_G(v)$ el grado de v en G (es decir, el número de arcos incidentes en v en el grafo G). A partir del grafo $G=(V,E)$ construimos el multigrafo $G'=(V',E')$ donde:

- V' corresponde a un nodo especial $v^*\notin V$ más todos los nodos en $v\in V$ no aislados (es decir, tales que $\text{grado}_G(v)\neq 0$), y
- E' contiene todos los elementos de E más $\text{grado}_G(v)-1$ arcos paralelos desde $v\in V$ a v^*, para todo $v\in V$ no aislado en G.

Observar que por construcción $\text{grado}_{G'}(v)=2\text{grado}_G(v)-1$, cualquiera sea $v\in V(G)$ no aislado. La instancia $f(\langle G,k\rangle)$ será $\langle G',2|E(G)|-k\rangle$. Dado $\langle G,k\rangle$ generar $f(\langle G,k\rangle)$ corresponde esencialmente a determinar el grado de cada nodo en G, lo que se puede implementar eficientemente en el modelo RAM en $O(|V(G)|+$

$|E(G)|$) pasos (asumiendo que el grafo G se codifica como una lista de adyacencia), luego f es calculable en tiempo polinomial por una máquina de Turing.

Veamos ahora que $\langle G,k\rangle \in$ VC si y solo si $\langle G',k'\rangle = f(\langle G,k\rangle) \in$ MaxMCut. En efecto, supongamos que $R \subseteq V(G)$ es un recubrimiento de nodos de G de tamaño a lo más k. Claramente, podemos asumir que R no contiene nodos aislados de G. Observar que

$$\begin{aligned}|\delta_{G'}(R)| &= |\delta_G(R)| + \left|\left\{vv^* \in E(G') : v \in R\right\}\right| = |\delta_G(R)| + \sum_{v\in R}(\text{grado}_G(v)-1)\\ &= \sum_{v\in V(G)} \text{grado}_G(v) - |R| = 2\,|E(G)| - |R|\,,\end{aligned}$$

donde la primera igualdad es por construcción de G' y porque $v^* \notin R$, la segunda igualdad es obvia, la tercera igualdad se tiene del hecho que no existen arcos en G entre los nodos de R (porque R es recubrimiento) y porque $\delta_G(R) = \delta_G(V(G)\setminus R)$, y la última igualdad se tiene del hecho que la suma de los grados de los nodos de un grafo es igual a dos veces el número de sus arcos. Por lo tanto, como $|R| \leq k$, sigue que R es un corte en G' de tamaño al menos $k' = 2|E(G)| - k$. Equivalentemente, $\langle G',k'\rangle = f(\langle G,k\rangle) \in$ MaxMCut. Supongamos ahora que $\langle G',k'\rangle = f(\langle G,k\rangle) \in$ MaxMCut. Se tiene entonces que existe $S \subseteq V(G')$ tal que $|\delta_{G'}(S)| \geq k' = 2|E(G)| - k$. Sea S uno de los conjuntos en que se alcanza el máximo de $|\delta_{G'}(S)|$. Observar que $\delta_{G'}(S) = \delta_{G'}(V(G')\setminus S)$. Sea $R \subseteq V(G)$ igual al conjunto S o $V(G')\setminus S$ que no contiene a v^*. Luego, considerando la identidad establecida más arriba y que $|\delta_{G'}(R)| \geq k' = 2|E(G)| - k$, sigue que $|R| \leq k$. Afirmamos que R es un recubrimiento de G, puesto que si no lo fuese existiría un arco $e \in E(G)$ con ambos extremos fuera de R. Afirmamos que agregando cualquiera de los extremos de e a R, digamos u, incrementaría el tamaño de $\delta_{G'}(R)$. En efecto, al menos $\text{grado}_G(u)$ extremos de los arcos incidentes en u tienen su otro extremo en el complemento de R (de hecho, e y $\text{grado}_G(u)-1$ arcos que inciden en v^*). Como $\text{grado}_{G'}(v) = 2\text{grado}_G(v) - 1$, sigue que al agregar u a R, la cardinalidad de $\delta_G(R)$ se incrementa en al menos 1. Sigue que R es un recubrimiento de vértices de G de tamaño a lo más k. En resumen, hemos establecido que f es una reducción a tiempo polinomial de VC a MaxMCut.

(iii).- Veamos primero que SetCover es NP–duro. Para ello veremos que VC $\leq_P$ SetCover. En efecto, dado $\langle G,k\rangle$ con G grafo y $k \in \mathbb{N}$, sea S_v el conjunto de vecinos de $v \in V(G)$ en G, i.e. $S_v = \{u \in V(G) : vu \in E(G)\}$. Afirmamos que $\langle G,k\rangle \in VC$ si y sólo si $\langle \{S_v : v \in V\}, k\rangle \in$ SetCover. En efecto, basta observar que $I \subseteq V(G)$ es un recubrimiento de nodos de G si y sólo si $\{S_i : i \in I\}$ es un recubrimiento de $\cup_{v\in V} S_v = E(G)$.

Dado $\langle G,k\rangle$ generar los S_v's corresponde esencialmente a determinar los arcos incidentes en cada nodo de G, lo que se puede implementar eficientemente en el modelo RAM en $O(|V(G)|+|E(G)|)$ pasos (asumiendo que el grafo G se codifica como una lista de adyacencia), luego la reducción es calculable en tiempo polinomial por una máquina de Turing.

En resumen, hemos establecido que VC $\leq_P$ SetCover.

Solo falta comprobar que SetCover está en NP. Para ello basta considerar el certificado $I \subseteq \{1,\ldots,n\}$ y verificar que $|I| \leq k$ y que los S_i's con $i \in I$ recubren $\Omega = \cup_{i=1}^n S_i$, lo que puede implementarse con facilidad en el modelo RAM en tiempo $O(n|\Omega|)$ que es a lo más cuadrático en el largo de la instancia considerada.

PROBLEMA 2:

(i).- Observar primero que si $L_1, L_2 \in$ NP$\cap$coNP, entonces L_1, L_2, L_1^c, y L_2^c están tanto en NP como en coNP.

Por definición de diferencia simétrica

$$L_1 \Delta L_2 = (L_1 \setminus L_2) \cup (L_2 \setminus L_1) = (L_1 \cap L_2^c) \cup (L_2 \cap L_1^c)\,.$$

Sigue que para probar que $L_1 \Delta L_2 \in$ NP basta con probar que NP es cerrado bajo unión e intersección.

Por otro lado,

$$(L_1 \Delta L_2)^c = (L_1^c \cup L_2) \cap (L_2^c \cup L_1)\,.$$

Sigue que para probar que $L_1 \Delta L_2 \in$ coNP nuevamente sólo se requiere observar que NP es cerrado bajo unión e intersección.

Hay varias formas de establecer que NP es cerrado bajo unión e intersección. Discutiremos una basada en la caracterización de NP en base a verificadores a tiempo polinomial. Sean $L', L'' \in$ NP. Por la caracterización vista en clases de los lenguajes en NP, sigue que existen verificadores V' y V'' a tiempo polinomial para L' y L'', respectivamente. Es decir:

$$\omega \in L' \iff \exists \pi'_\omega, \text{ tal que } V' \text{ acepta } (\omega, \pi'_\omega)\,,$$
$$\omega \in L'' \iff \exists \pi''_\omega, \text{ tal que } V'' \text{ acepta } (\omega, \pi''_\omega)\,.$$

Sea $V_\cap$ (respectivamente $V_\cup$) tal que acepta $(\omega, \pi'_\omega, \pi''_\omega)$ si y solo si V' acepta (ω, π'_ω) y (respectivamente, o) V'' acepta (ω, π''_ω). Es fácil ver que $V_\cap$ (respectivamente, $V_\cup$) es un verificador a tiempo polinomial de $L' \cap L''$ (respectivamente $L' \cup L''$). En resumen, NP es cerrado bajo intersección y unión.

(ii).- Sea L decidido por una máquina de Turing universalment no-determinista M a tiempo polinomial $p(\cdot)$. Afirmamos que $L \in$ coNL, o equivalentement que $\overline{L} \in$ NP. En efecto, observar que $\omega \in L$ si y solo si M en la entrada ω acepta cualquiera que sea el contenido ρ de su cinta no-determinista. Sea $\overline{M}$ la máquina de Turing que simula M pero que acepta (respectivamente, rechaza) si M rechaza (respectivamente, acepta). Observar que $\omega \in \overline{L}$ si y solo si existe un ρ contenido de la cinta no-determinista de M que hace que esta rechace ω. Equivalentemente, $\overline{M}$ decide $\overline{L}$. Como $\overline{M}$ es una máquina de Turing no-determinista a tiempo $p(\cdot)$, concluimos que $\overline{L} \in$ NP.

El argumento del párrafo anterior es fácil de revertir, lo que permite concluir que si $L \in$ coNP, entonces es decidible en tiempo polinomial por una máquina de Turing universalmente no-determinista.

En resumen, un lenguaje está en coNP si y sólo si puede ser decidido por una máquina de Turing universalmente no-determinista en tiempo polinomial.

(iii).- La idea es simular una máquina no–determinista por una probabilista reemplazando los bit no–deterministas por bits aleatorios, pero de manera que inclusive la existencia de una única rama de cálculo no–determinista de aceptación lleve a que la máquina probabilista acepte con una probabilidad mayor que $1/2$. Para garantizar esto, la máquina probabilista partirá (esencialmente) lanzando una moneda y aceptando inmediatamente si el resultado es cara, independiente de cual sea la entrada. Si el resultado es sello, entonces la máquina probabilista simulará la máquina no–determinista como se describió más arriba.

Formalmente, consideremos $L \in$ NP y M una mTND a tiempo polinomial $p(n)$ que decide L. Sin pérdida de generalidad asumimos que en cada una de sus transiciones M opta entre dos posibles transiciones no deterministas (ambas pudiendo llevar a las misma configuración). Definimos la mT probabilista M' que en la entrada ω realiza los siguientes pasos:

1. Lanza $p(|\omega|)+1$ monedas. Si el primer lanzamiento sale cara y alguno de los lanzamientos sala sello, entonces para y acepta.

2. Si la máquina no para en el paso anterior, entonces simula M en la entrada ω, pero cada vez que M requiere un bit no determinista, se lanza una moneda y se prosigue la simulación utilizando el resultado de dicho lanzamiento como el bit no–determinista requerido por M. Si M eventualmente acepta, entonces se acepta.

Observar que M' acepta en el paso (1) con probabilidad $1/2(1-2^{-p(|\omega|)})$. En el paso (2) la máquina M' acepta con probabilidad 0 si $\omega \notin L$. Luego, la probabilidad que M' acepta cuando $\omega \notin L$ es $1/2(1-2^{-p(|\omega|)}) < 1/2$. En el paso (2) la máquina M' acepta con probabilidad al menos $1/2^{p(|\omega|)}$ si $\omega \in L$, puesto que M debe aceptar ω en al menos una de sus $2^{p(|\omega|)}$ ramas de cálculo no–determinista, y M' simula dicha rama con probabilidad al menos $1/2^{p(|\omega|)}$. Luego, la probabilidad que M' acepta cuando $\omega \in L$ es al menos $1/2(1-2^{-p(|\omega|)}) + 1/2^{p(|\omega|)} > 1/2$.

(iv).- Supongamos que P = NP. Sigue que P = coNP. Sea L el conjunto de las (codificaciones de) las fórmulas Booleanas ψ tales que para todo $y \in \{0,1\}^m$ se tiene que $\psi(y) = 1$, donde ψ es sobre m variables. Es fácil ver que $L \in$ coNP. Luego, por hipótesis, se tiene que $L \in$ P. Sea V máquina de Turing que decide L en tiempo polinomial. Sigue que $\langle \psi \rangle$, donde ψ es como en el enunciado, está en Σ_2SAT si y solo si existe $x \in \{0,1\}^n$ tal que V acepta $\langle \varphi(x,\cdot) \rangle$. En otras palabras, existe un verificador a tiempo polinomial para Σ_2SAT. Por caracterización vista de los lenguajes en NP, lo anterior equivale a decir que Σ_2SAT está en NP. Nuevamente, por hipótesis, sigue que Σ_2SAT $\in$ P como se quería establecer.