Ma50b Complejidad Computacional **01 de Julio de 2010**

Pauta Control 3

Prof. Cátedra: M. Kiwi *Prof. Auxiliar: R. Bricenño*

PROBLEMA 1:

(i.1).- Para demostrar que DAPATH es NL-duro, basta usar el mismo argumento utilizado para demostrar que PATH es NL-duro pero considerando máquinas de Turing aumentadas con relojes.

(i.2).- Definamos el lenguaje DACICL como la colección de $\langle G,s,t\rangle$ tales que G es un digrafo acíclico. Observar que DAPATH = PATH $\cap$ DACICL. Sabemos que PATH está en NL y es fácil ver que NL es cerrado bajo intersección. Para obtener la conclusión deseada, basta entonces demostrar que DACICL está en NL. Pero NL = coNL, por lo que es suficiente probar que DACICL está en coNL. Consideremos la máquina de Turing fuera de línea,

M = En $\langle G,s,t\rangle$,
(1).- De manera no-determinista, adivinar universalmente, un nodo $s \in V(G)$.
(2).- Hacer $v \leftarrow s$.
(3).- Para $i \in \{1,\ldots,|V(G)|\}$
(4).- De manera no-determinista, adivinar universalmente, un nodo $v' \in V(G)$.
(5).- Si $vv' \notin E(G)$, entonces *Rechazar*.
(6).- Si $v' = s$, entonces *Rechazar*.
(7).- Hacer $v \leftarrow v'$.
(8).- *Aceptar*.

Dado que M sólo requiere almacenar los valores de $s,v,v' \in V(G)$ e $i \in \{1,\ldots,|V(G)|\}$, se tiene que M ocupa $O(\log|V(G)|)$ espacio, i.e. M es una máquina de Turing tipo coNL.

Por otro lado, si G no es acíclico, entonces posee un ciclo de largo a lo más $|V(G)|$. En este caso, M hara una secuencia de adivinanzas de nodos partiendo con un nodo en el cíclo y recorriendo secuencialmente el ciclo hasta volver al nodo inicial, en cuyo instante M rechazará. Si G es acíclico, es fácil ver que M no rechazará en los pasos (5) y (6), por lo que eventualmente aceptará. Sigue que M reconoce DACICL, y por lo tanto DACICL $\in$ coNL = NL como se quería demostrar.

(ii).- Para establecer el resultado, veremos que CIRC$-$VAL log-espacio reduce a LP. Sea $\langle C,\vec{a}\rangle$ tal que C es un cicuito Booleano en n entradas y $\vec{a} = (a_1,\ldots,a_n) \in \{0,1\}^n$. Supongamos que V es el conjunto de puertas lógicas de C, que $V' = \{v'_1,\ldots,v'_n\}$ es el conjunto de n entradas, y que $o \in V$ es la salida del circuito. Constuiremos una intancia de LP con variables x_v donde $v \in V \cup V'$. Las desigualdades a considerar son:

- Para todo $i \in \{1,\ldots,n\}$, $x_{v'_i} = a_i$.
- Para todo $v \in V$, $0 \leq x_v \leq 1$.

- Para todo $v \in V$ puerta lógica de negación con u la puerta lógica que alimenta v,

$$x_v = 1 - x_u .$$

- Para todo $v \in V$ puerta lógica de conjunción con $u_1, \ldots, u_{k_v}$ puertas lógicas que alimentan a v,

$$\begin{aligned} x_{u_j} &\geq x_v , \qquad \text{para todo } i \in \{1, \ldots, k_v\} , \\ x_v &\geq \left(\sum_{j=1}^{k_v} x_{u_j} \right) - (k_v - 1) . \end{aligned}$$

- Para todo $v \in V$ puerta lógica de disyunción con $u_1, \ldots, u_{k_v}$ puertas lógicas que alimentan a v,

$$\begin{aligned} x_v &\geq x_{u_j} , \qquad \text{para todo } i \in \{1, \ldots, k_v\} , \\ x_v &\leq \sum_{j=1}^{k_v} x_{u_j} . \end{aligned}$$

Finalmente, consideramos la función lineal a optimizar como x_o. Resumiendo, a la instancia $\langle C, \vec{a} \rangle$ de CIRC$-$VAL le hemos asociado una mátriz A a coordenadas enteras y vectores b y c también a coordenadas enteras.

Afirmamos que $\langle C, \vec{a} \rangle \in$ CIRC$-$VAL si y solo si $\langle A, b, c, 1 \rangle \in$ LP. En efecto, supongamos que $\langle C, \vec{a} \rangle \in$ CIRC$-$VAL. Definimos $x_v \in \{0,1\}$ como el valor que toma la puerta lógica v de C cuando lo evaluamos en $\vec{a}$, y hacemos $x_{v'_i} = a_i$ para cada v'_i entrada de C. Es fácil ver que $x = (x_v)_{v \in V \cup V'}$ es tal que $Ax \leq b$ y $c^T x = x_o = 1$. Luego, $\langle A, b, c, 1 \rangle \in$ LP. Por otro lado, si $\langle A, b, c, 1 \rangle \in$ LP, sigue que existe $x = (x_v)_{v \in V \cup V'}$ que satisface $Ax \leq b$ y tal que $x_o \geq 1$. No es difícil verificar que el valor x_v corresponde al valor de verdad que toma la puerta v del circuito C evaluado en $\vec{a}$. Sigue que $C(\vec{a}) = x_o \geq 1$, es decir $\langle C, \vec{a} \rangle \in$ CIRC$-$VAL.

La característica local de la reducción que le asocia $\langle A, b, c, 1 \rangle$ a $\langle C, \vec{a} \rangle$ permite que la reducción pueda ser calculada por una máquina a log-espacio. En resumen, $CIRCVAL \leq_L$ LP por lo que LP es NL-duro.

PROBLEMA 2:

(i).- Sean $C(\cdot, \cdot)$ y $\mathcal{S}_C$ como en el enunciado. Afirmamos que $\mathrm{VC}(\mathcal{S}_C) \leq \ell$. Por contradicción, supongamos que existe un X de cardinal $\ell' > \ell$. Sigue que

$$2^{\ell'} = \left| 2^X \right| = \left| \left\{ S_\alpha \cap X : \alpha \in \{0,1\}^\ell \right\} \right| \leq 2^\ell ,$$

i.e. $\ell' \leq \ell$, contradicción. Esto concluye la demostración de la afirmación.

Sigue que $\omega = \langle C(\cdot, \cdot), k \rangle \in$ VC$-$DIM si y solo si

$$(k \leq \ell) \wedge \left(\exists x_1, \ldots, x_k \in \{0,1\}^m, \forall I \subseteq \{1, \ldots, k\}, \exists \alpha \in \{0,1\}^\ell, (\forall i \in \{1, \ldots, k\}, C(\alpha, x_i) = 1 \Leftrightarrow i \in I) \right) . \quad (1)$$

Notar que dado $\omega = \langle C(\cdot, \cdot), k \rangle$ tal que $k \leq \ell$, la afirmación $\forall i \in \{1, \ldots, k\}, C(\alpha, x_i) = 1 \Leftrightarrow i \in I$ puede ser decidida en tiempo polinomial en $|\omega|$ (dado que $\ell \leq |\omega|$ y que la evaluación de $C(\cdot, \cdot)$ puede ser realizada en tiempo polinomial en $|\omega|$). Sigue facilmente que el valor de verdad de la expresión en (1) puede ser decidida por una máquina de Turing alternante del tipo Σ_3^{P}.

(ii).- Bastará verificar que una asignación $\vec{a} = (a_1, \ldots, a_n)$ elegida al azar uniformemente en $\{0,1\}^n$ tiene una probabilidad positiva de satisfacer una fórmula Booleana φ con las características del enunciado. En efecto, sea $\varphi = \wedge_{i=1}^{m} C_i(x_1, \ldots, x_n)$ donde cada C_i es un claúsula con $k_i \geq t \log_2 n$ literales distintos. Notar que por desigualdad de Boole,

$$\begin{aligned} \mathbb{P}_{\vec{a} \in_R \{0,1\}^n} (\varphi(a_1, \ldots, a_n) = 0) &= \mathbb{P}_{\vec{a} \in_R \{0,1\}^n} (\exists i \in \{1, \ldots, m\}, C_i(a_1, \ldots, a_n) = 0) \\ &\leq \sum_{i=1}^{m} \mathbb{P}_{\vec{a} \in_R \{0,1\}^n} (C_i(a_1, \ldots, a_n) = 0) \, . \end{aligned}$$

Pero, $C_i(a_1, \ldots, a_n) = 0$ si y solo si cada uno de los literales que aparecen en la claúsula C_i evaluan a 0 en $(a_1, \ldots, a_n)$. Esto ocurre con probabilidad a lo más $2^{-k_i} \leq n^{-t}$ cuando $(a_1, \ldots, a_n)$ está elegido al azar uniformemente en $\{0,1\}^n$. Como $m < n^t$, sigue que

$$\mathbb{P}_{\vec{a} \in_R \{0,1\}^n} (\varphi(a_1, \ldots, a_n) = 0) \leq \frac{m}{n^t} < 1 \, .$$

Luego, $\mathbb{P}_{\vec{a} \in_R \{0,1\}^n} (\varphi(a_1, \ldots, a_n) = 1) > 0$, i.e. existe $(a_1, \ldots, a_n) \in \{0,1\}^n$ tal que $\varphi(a_1, \ldots, a_n) = 1$.

PROBLEMA 3:

(i).- Sea $L \in$ BPP decidido por una máquina de Turing probabilista R a tiempo polinomial $q(\cdot)$ tal que

$$\begin{aligned} \omega \in L &\implies \mathbb{P}_{\rho \in_R \{0,1\}^{q(|\omega|)}} (R(\omega, \rho) = 1) \geq \frac{2}{3} \, , \\ \omega \notin L &\implies \mathbb{P}_{\rho \in_R \{0,1\}^{q(|\omega|)}} (R(\omega, \rho) = 1) \leq \frac{1}{3} \, . \end{aligned}$$

Sea M la maąquina de Turing a tiempo polinomial $q'(\cdot)$ como la del enunciado cuya existencia está garantizada por la del generador de bits pseudo-aleatorio. Sea $\varepsilon > 0$ y c suficientemente grande tal que si $p(n) = (n+2)^c$ se tiene que $p(n^\varepsilon) \geq q(n)$ para todo $n \in \mathbb{N}$. Sin pérdida de generalidad podemos modificar R y asumir que usa exactamente $p(n^\varepsilon)$ bits aleatorios en todas las entradas de largo n. Consideremos la siguiente máquina de Turing:

D = En ω,
(1).- Hacer $n \leftarrow |\omega|$.
(2).- Hacer $cont \leftarrow 0$.
(3).- For $\rho \in \{0,1\}^{n^\varepsilon}$
(4).- Simular M en $\langle p(\cdot), \rho \rangle$ y obtener $\widetilde{\rho} = G_{n^\varepsilon, p(\cdot)}(\rho)$.
(5).- Simular R en ω utilizando $\widetilde{\rho}$ en vez de bits aleatorios.
(6).- Si R acepta, hacer $cont \leftarrow cont + 1$.
(7).- Si $cont / 2^{q(n)} > 1/2$, entonces *Aceptar*, de lo contrario *Rechazar*.

Observar que D es a tiempo $O(2^{n^\varepsilon}(q'(n^\varepsilon) + q(n))) = O(2^{n^\varepsilon})$.

Veamos que D decide L. Por resultado visto, se tiene que existe una familia de circuitos Booleanos log-espacio uniforme $(C_n)_{n \in \mathbb{N}}$ tal que C_n actúa en $n + p(n^\varepsilon)$ entradas y si $n = |\omega|$, entonces

$$R(\omega, \rho) = 1 \Longleftrightarrow C_n(\omega, \rho) = 1 \, .$$

Además, existe un polinomio $q''(\cdot)$ tal que $|C_n| \leq q''(\cdot)$. Sea n_0 suficientemente grande tal que para todo $n \geq n_0$ se tiene que $S(n) > \text{máx}\{q''(n), 6\}$ (notar que n_0 existe dado que $S(n) > n^{\omega(1)}$). Sigue que si ω es una instancia de L de largo $n \geq n_0$,

$$\begin{aligned}
&\left|\mathbb{P}_{\rho \in_R \{0,1\}^{n^\varepsilon}}\left(R(\omega, G_{n^\varepsilon,p(\cdot)}(\rho)) = 1\right) - \mathbb{P}_{\rho'' \in_R \{0,1\}^{p(n^\varepsilon)}}\left(R(\omega,\rho'') = 1\right)\right| \\
&= \left|\mathbb{P}_{\rho \in_R \{0,1\}^{n^\varepsilon}}\left(C_n(\omega, G_{n^\varepsilon,p(\cdot)}(\rho)) = 1\right) - \mathbb{P}_{\rho'' \in_R \{0,1\}^{p(n^\varepsilon)}}\left(R(\omega,\rho'') = 1\right)\right| \\
&\leq \left|\mathbb{P}_{\rho' \in_R \{0,1\}^{p(n^\varepsilon)}}\left(C_n(\omega,\rho') = 1\right) - \mathbb{P}_{\rho'' \in_R \{0,1\}^{p(n^\varepsilon)}}\left(R(\omega,\rho'') = 1\right)\right| + \frac{1}{S(n)} \\
&= \frac{1}{S(n)}\,.
\end{aligned}$$

Como $S(n) \geq 6$, sigue que $P_{ac} = \mathbb{P}_{\rho \in_R \{0,1\}^{n^\varepsilon}}\left(R(\omega, G_{n^\varepsilon,p(\cdot)}(\rho)) = 1\right)$ es mayor que $1/2$ si $\omega \in L$, y menor que $1/2$ si $\omega \notin L$. Pero en el paso (7), el valor $cont/2^{q(n)}$ que calcula la máquina de Turing D es justamente P_{ac}. Se concluye que D decide L. Sigue que para todo $\varepsilon > 0$, se tiene que $L \in \text{DTIEMPO}(2^{n^\varepsilon})$, luego $L \in \cap_{\varepsilon>0}\text{DTIEMPO}(2^{n^\varepsilon})$.

(ii).- Sean M y $G_{n,p(\cdot)}$ como en el enunciado. Fijamos el polinomio $p(n) = 2n$ y definimos $L_{p(\cdot)} = \cup_n G_{n,p(\cdot)}(\{0,1\}^n)$. Veremos que $L_{p(\cdot)} \in \text{NP} \setminus \text{P}$.

Para probar que $L_{p(\cdot)} \in \text{NP}$, consideremos la siguiente máquina de Turing no-determinista

N = En ω,
(1).- Si $n = |\omega|$ es impar, entonces *Rechazar*.
(2).- De manera no-determinista adivina $\rho \in \{0,1\}^{n/2}$.
(3).- Simula M en la entrada $\langle p(\cdot), \rho\rangle$ y obtiene $\sigma = G_{n,p(\cdot)}(\rho)$.
(4).- Si $\sigma = \omega$, entonces *Aceptar*. En caso contrario, *Rechazar*.

Dado que M es a tiempo polinomial, sigue facilmente que N es a tiempo no-determinista polinomial. Observar además, que N acepta ω si y solo si existe $n = |\omega|/2 \in \mathbb{N}$ y $\rho \in \{0,1\}^n$ tal que $\omega = G_{n,p(\cdot)}(\rho)$. Es decir, N acepta L. Resumiendo, $L_{p(\cdot)} \in \text{NP}$.

Para efectos de obtener una contradicción, supongamos ahora que $L_{p(\cdot)} \in \text{P}$. Por resultado visto, existe una familia de circuitos Booleanos log-espacio uniforme $(C_n)_{n\in\mathbb{N}}$ tal que para todo $\omega \in \{0,1\}^n$,

$$\omega \in L_{p(\cdot)} \Longleftrightarrow C_n(\omega) = 1\,.$$

Además, existe un polinomio $q(\cdot)$ tal que $|C_n| \leq q(n)$ cualquiera sea $n \in \mathbb{N}$.

Observar que

$$\mathbb{P}_{x \in_R \{0,1\}^n}\left(C(G_{n,p(\cdot)}(x)) = 1\right) = 1\,.$$

Por otro lado,

$$\mathbb{P}_{y \in_R \{0,1\}^{2n}}\left(C(y) = 1\right) = \mathbb{P}_{y \in_R \{0,1\}^{2n}}\left(y \in L_{p(\cdot)}\right) = \frac{\left|\left\{G_{n,p(\cdot)}(x) : x \in \{0,1\}^n\right\}\right|}{2^{2n}} \leq \frac{2^n}{2^{2n}} = \frac{1}{2^n}\,.$$

Sea $n \geq 1$ suficientemente grande tal que $S(n) > \text{máx}\{2, q(n)\}$ (observar que n existe porque $S(n) > n^{\omega(1)}$). Sigue que,

$$\left|\mathbb{P}_{x \in_R \{0,1\}^n}\left(C(G_{n,p(\cdot)}(x)) = 1\right) - \mathbb{P}_{y \in_R \{0,1\}^{p(n)}}\left(C(y) = 1\right)\right| \geq 1 - \frac{1}{2^n} \geq \frac{1}{2} > \frac{1}{S(n)}\,,$$

contradiciendo la existencia del generador de bits pseudo-aleatorio criptográficamente seguro.