Ma5201 Complejidad Computacional **20 de Julio de 2013**

Pauta Control 3

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

PROBLEMA 1:

(i).- Veamos primero que CicloImpar esta en NL. Para ello consideremos el siguiente algoritmo $\mathcal{A}$:

input: $\langle G=(V,E)\rangle$ instancia de CicloImpar.
No-deterministamente adivina $1 \leq \ell \leq |V|$ número impar;
No-deterministamente adivina $s \in V$;
$v \leftarrow s$;
for $i=1,\ldots,\ell$ **do**
  No-deterministamente adivina $u \in V$;
  if $vu \notin E$ **then** return(Rechazar);
if $v=s$ **then**
  return(Aceptar);
else
  return(Rechazar);

Claramente, $\mathcal{A}$ acepta si y sólo si $\langle G=(V,E)\rangle \in$ CicloImpar. Más aún, $\mathcal{A}$ sólo necesita almacenar ℓ, s, v, y u, requiriendo la representación de cada uno de ellos espacio $O(\log|V|)$. Luego, $\mathcal{A}$ es un algoritmo no-determinista a espacio logarítmico que decide CicloImpar.

Veamos ahora que CicloImpar es NL-duro. Bastará probar que PATH $\leq_L$ CicloImpar. La reducción le asociará a $\langle G=(V,E),s,t\rangle$ instancia de PATH un digrafo G' que se obtiene a partir de G reemplazando cada uno de sus arcos $e=uv$ por un camino de u a v de largo 2, y agregando un arco de t a s. Formalmente, $G'=(V',E')$ donde $V'=V\cup E$ y

$$E' = \bigcup_{e=uv\in E} \{ue, ev\} \cup \{ts\}.$$

Es fácil ver que la reducción puede implementarse en log-espacio (dado que esencialmente consiste en generar dos arcos por cada arco de G). Además, si $s=v_0,v_1,\ldots v_\ell=t$ es un st-dicamino en G, entonces $s=v_0,v_0v_1,v_1,v_1v_2,\ldots,v_\ell=t,v_0=s$ es un cíclo de largo impar en G'. Por otro lado, si G' posee un cíclo impar, entonces dicho cíclo necesariamente debe ocupar el arco ts, por lo que la parte del cíclo que va de s a t es un dicamino de s a t en G'. Tomando un nodo por medio de este último dicamino, se obtiene un st-dicamino en G, i.e. $\langle G,s,t\rangle \in$ CicloImpar.

Resumiendo, PATH log-espacio reduce a CicloImpar.

(ii).- Por resultado visto, $\Sigma_k^{\mathrm{P}} = \mathrm{NP}^{\Sigma_{k-1}^{\mathrm{P}}}$. Como $\Sigma_{k-1}^{\mathrm{P}} \subseteq \Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}}$, el mismo argumento usado para probar la primera inclusión de la parte (ii), nos permite concluir que $\Sigma_k^{\mathrm{P}} \subseteq \mathrm{NP}^{\Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}}}$.

Consideremos ahora L decidido por un máquina de Turing no-determinista N a tiempo polinomial $p(\cdot)$ con acceso a un oráculo $O \in \Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}}$. En particular, por definición de Σ_k^{P} y de Π_k^{P}, se tiene que existen máquinas de Turing a tiempo polinomial M y $\overline{M}$, y polinomios $p(\cdot)$ y $\overline{p}(\cdot)$, tales que

$$\begin{aligned}
\sigma \in O &\iff \exists u_1 \in \{0,1\}^{q(|\sigma|)}, \forall u_2 \in \{0,1\}^{q(|\sigma|)}, \ldots, Q_k u_k \in \{0,1\}^{q(|\sigma|)}, M(\sigma, u_1, \ldots, u_k) = \text{Aceptar}, \\
\sigma \in O &\iff \forall u_1 \in \{0,1\}^{q(|\sigma|)}, \exists u_2 \in \{0,1\}^{q(|\sigma|)}, \ldots, \overline{Q}_k u_k \in \{0,1\}^{q(|\sigma|)}, \overline{M}(\sigma, u_1, \ldots, u_k) = \text{Aceptar},
\end{aligned}$$

done Q_k denota $\exists$ o $\forall$ si i es par o impar, respectivamente.

Notar entonces que $\omega \in L$ si y sólo si existe una secuencia de elecciones no-deterministas $c_1, \ldots, c_s \in \{0,1\}$ y respuestas $a_1, \ldots, a_t \in \{0,1\}$ (donde s y t son el máximo número de elecciones no-deterministas y consultas al oráculo que que realiza N en entradas de largo $|\omega|$ – en particular $s, t \leq p(|\omega|)$), tal que en la entrada ω si la máquina N usa las elecciones $c_1, \ldots, c_s$ en su ejecución y recibe a_i como respuesta a su i-ésima consulta, entonces: (1) N alcanza un estado de aceptación, y (2) todas las respuestas recibidas son correctas. Sea σ_i la i-ésima consulta que N realiza a su oráculo en la entrada ω usando la elecciones no-deterministas $c_1, \ldots, c_s$, y obteniendo como respuestas a sus consultas previas $a_1, \ldots, a_{i-1}$. La condición (2) anterior puede reformularse de la siguiente forma: Si $a_i = 1$, entonces $\exists u_1, \forall u_2, \ldots, Q_k u_k$ tal que $M(\sigma_i, u_1, \ldots, u_k) = \text{Aceptar}$, y si $a_i = 0$, entonces $\exists u_1, \forall u_2, \ldots, Q_k u_k$ tal que $\overline{M}(\sigma_i, u_1, \ldots, u_k) \neq \text{Aceptar}$. Sigue que,

$$\begin{aligned}
\omega \in L \iff & \exists c_1, \ldots, c_s, a_1, \ldots, a_t, u_1^{(1)}, \ldots, u_t^{(1)}, \forall u_1^{(2)}, \ldots, u_t^{(2)}, \ldots Q_k u_1^{(k)}, \ldots, u_t^{(k)}, \text{ tal que } N \text{ acepta } \omega \\
& \text{en las elecciones no-deterministas } c_1, \ldots, c_s \text{ y respuestas } a_1, \ldots, a_t, \text{ y para todo } i \in [t], \\
& \qquad a_i = 1 \implies M(\sigma_i, u_i^{(1)}, \ldots, u_i^{(k)}) = \text{Aceptar}, \\
& \qquad a_i = 0 \implies \overline{M}(\sigma_i, u_i^{(1)}, \ldots, u_i^{(k)}) \neq \text{Aceptar}.
\end{aligned}$$

Verificando que la parte derecha de la equivalencia anterior es una afirmación del tipo Σ_k^{P}, sigue que $L \in \Sigma_k^{\mathrm{P}}$.

PROBLEMA 2:

(i).- Sea $\langle G = (V, E), s, t \rangle$ una instancia de UPATH. Denotemos por d_v el grado del nodo v en G. Definimos G' el multigrafo que se obtiene al agregar a cada nodo del grafo G exactamente $|V| - d_v$ búcles. Claramente se tiene que G' es multi-grafo $|V|$-regular. Como $d_v \leq |V| - 1$, se tiene que en la construcción de G' a partir de G, a cada nodo se le agrega al menos un búcle. Además, como las componentes conexas de G' y G son idénticas, se tiene que $\langle G, s, t \rangle \in$ UPATH si y sólo si $\langle G', s, t \rangle \in$ RegUPATH.

Lo único que queda por establecer es que la reducción que a $\langle G = (V, E) \rangle$ le asocia $\langle G' = (V, E') \rangle$ es a log-espacio. [FALTA]

(ii).- Sea G una instancia de RegUPATH y d tal que G es d-regular, y sean N_G y N_H el número de nodos de G y H, respectivamente. Notar que H también es d-regular y que $d \leq N_G$. Sea $\widehat{B} = (1/d)A(H)$, y $e_s \in \mathbb{R}_+^{|N_H|}$ tal que todas sus coordenadas son iguales a 0 salvo la coordenada s. La observación clave es que la probabilidad que una caminata aleatoria en G de ℓ pasos que comienza en s termine en $u \in H$ esta dada por la coordenada u del vector $\widehat{B}^\ell e_s$. Luego, dado que $1 - x \leq e^{-x}$, tomando $\ell = \lfloor 8N_G^4 \ln(2N_G) \rfloor$, recordando que $d, N_H \leq N_G$, y asumiendo el resultado del enunciado, para todo nodo u de H, se tiene que

$$\left| \left(\widehat{B}^\ell e_s \right)_u - \frac{1}{N_H} \right| \leq \left\| \widehat{B}^\ell p - \frac{1}{N_H} \mathbb{I} \right\|_2 \leq \left(1 - \frac{1}{8dN_H^3} \right)^\ell \leq \exp\left(-\frac{N_G^3}{dN_H^3} \ln(2N_G) \right) \leq e^{-\ln(2N_G)} = \frac{1}{2N_G} \leq \frac{1}{2N_H}.$$

Sigue que $\left(\widehat{B}^\ell e_s \right)_u \geq 1/(2N_H) \geq 1/(2N_G)$, como se quería establecer.

(iii).- Consideremos el siguiente algoritmo $\mathcal{A}$:

```
  input: ⟨G = (V,E), s, t⟩ instancia de RegUPATH.
1 ℓ ← ⌊8|V|^4 ln(2|V|)⌋;
2 m ← ⌊2|V| ln 3⌋;
3 for i = 1,...,m do
4     v ← s;
5     for j = 0,...,ℓ do
6         Elegir u ∈_R {w ∈ V : vw ∈ E};
7         v ← u;
8     if v = t then return(Aceptar);
9 return(Rechazar);
```

La ejecución de $\mathcal{A}$ en $\langle G=(V,E),s,t\rangle$ sólo requiere almacenar i, ℓ, v, u, y m, todos de tamaño $O(\log|V|)$, y además elegir (y almacenar) un índice tomado uniformemente al azar en $[d]$, donde d es el grado de regularidad de G (esto último se requiere para implementar el Paso 6 en la especificación de $\mathcal{A}$). Por lo tanto, $\mathcal{A}$ es a log-espacio. Por otro lado, la ejecución de $\mathcal{A}$ esta dominada por las $\ell\cdot m$ veces que se ejecutan los Pasos 6 y 7. Ambos pasos se pueden ejecutar en tiempo lineal en el tamaño de la entrada. Dado que $\ell\cdot m$ es polinomial en el tamaño de la entrada, sigue que $\mathcal{A}$ es a tiempo polinomial.

Claramente $\mathcal{A}$ sólo acepta $\langle G=(V,E),s,t\rangle$ si encuentra un camino de s a t. Luego, $\mathcal{A}$ nunca acepta si $\langle G=(V,E),s,t\rangle\notin$ RegUPATH. Supongamos entonces que $\langle G=(V,E),s,t\rangle\in$ RegUPATH. De la parte (ii) tenemos que en cada iteración del loop del Paso 3, el algoritmo acepta con probabilidad al menos $1/(2|V|)$. Luego, la probabilidad que no acepte en ninguna de las m veces que el loop se ejecuta es a lo más,

$$\left(1-\frac{1}{2|V|}\right)^m \leq e^{-m/(2|V|)} \leq \frac{1}{3}.$$

Sigue que $\mathcal{A}$ acepta $\langle G=(V,E),s,t\rangle\in$ RegUPATH con probabilidad al menos $2/3$. En resumen, hemos establecido que RegUPATH $\in$ RL.