**Ma50b Complejidad Computacional** **18 de Junio de 2005**

# Pauta Control 3

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

PROBLEMA 1:

(i).- Basta ver que $VAL{-}CIRC$ log–espacio reduce a $LP$. En efecto, sea $\langle C,a\rangle$ tal que $C$ es un circuito Booleano en $n$ variables $x=(x_1,\ldots,x_n)$ y sea $a=(a_1,\ldots,a_n)\in\{0,1\}^n$ una asignación de valores de verdad para dichas variables. Sin pérdida de generalidad supondremos que $C$ es un circuito Booleana con puertas lógicas $\vee$, $\wedge$ y $\neg$ (es fácil ver que podemos restringir $VAL{-}CIRC$ a circuitos de este tipo sin sacrificar la P–completitud). En lo que sigue, denotaremos las puertas lógicas de $C$ por $V(C)$ y consideraremos que las variables corresponden también a puertas lógicas pero de un valor constante dado por la asignación de valores de verdad $a$.

A cada variable $x_i$ le asociaremos una variable real $z_i$. A cada nodo $v$ del circuito $C$ le asociaremos una variable real $z_v$. Consideremos entonces el siguiente conjunto de desigualdades:

- $z_i=a_i$, para todo $i\in\{1,\ldots,n\}$ — lo que garantiza que la variable real $z_i$ asociada a una variable Booleana $x_i$ toma el valor de verdad correspondiente a $a_i$.
- si $v\in V(C)$ corresponde a una negación, entonces $z_v=1-z_u$ para el único nodo $u$ tal $uv$ es un arco del circuito $C$ — lo que garantizará que la variable real $z_v$ tome el valor 1 si y sólo si $z_u=0$.
- si $v\in V(C)$ corresponde a una disyunción y $u_1,\ldots u_d$ son los nodos desde los cuales llegan arcos a $v$, entonces $0\leq z_v\leq 1$, $z_v\geq z_{u_i}$ para todo $i=1,\ldots,d$, y $z_{u_1}+\ldots+z_{u_d}\geq z_v$ — lo que garantizará que la variable real $z_v$ tome el valor 1 si y sólo si algún $z_{u_i}$ toma el valor 1.
- si $v\in V(C)$ corresponde a una conjunción y $u_1,\ldots u_d$ son los nodos desde los cuales llegan arcos a $v$, entonces $0\leq z_v\leq 1$, $z_v\leq z_{u_i}$ para todo $i=1,\ldots,d$, y $z_v\geq 1-(1-z_{u_1})-\ldots-(1-z_{u_d})$ — lo que garantizará que la variable real $z_v$ tome el valor 1 si y sólo si todos los $z_{u_i}$'s toman el valor 1.

Claramente, el sistema lineal recién explicitado se puede poner en la forma $Az\leq b$ para algún $A\in\mathbb{Z}^{|V(C)|\times|V(C)|}$ y $b\in\mathbb{Z}^{|V(C)|}$. Si denotamos por $v^*$ la puerta lógica del circuito $C$ correspondiente a la salida de $C$, definimos $c=(c_v)_{v\in V(C)}$ de manera que $c_{v^*}=1$ y $c_v=0$ si $v\neq v^*$, y hacemos $B=1$, entonces no es difícil ver que $z_v$ toma el valor 1 o 0 dependiendo de si la puerta lógica asociada a $v\in V(C)$ toma el valor 1 o 0 cuando se evalúa el circuito en $C$ en la entrada $a$¿ En particular $C(a)=1$ si y sólo si $z_{v^*}=1$. En otras palabras,

$$\langle C,a\rangle\in VAL{-}CIRC\Longleftrightarrow\langle A,b,c;B\rangle\in\mathrm{LP}\,.$$

La característica altamente local de la construcción del sistema de desigualdades permite establecer, de manera relativamente sencilla, que la reducción es a espacio logarítmico.

(ii).- Como $co\mathrm{NL}=\mathrm{NL}$ bastará probar que $\overline{SAT}$ es NL–completo. Para ello, veremos que $CAMINO$ log–espacio reduce a $\overline{SAT}$. En efecto, sea $\langle G,s,t\rangle$ tal que $G=(V,E)$ es un digrafo y $s,t\in V$. A cada nodo $v$ del

grafo $G$ le asociamos una variable Booleana $x_v$. A cada arco $uv$ del grafo $G$ le asociamos una cláusula $x_u \Rightarrow x_v$ o equivalentemente $\overline{x_u} \vee x_v$. Sea entonces

$$\varphi = x_s \wedge \overline{x_t} \wedge_{uv \in E} \left(\overline{x_u} \vee x_v\right).$$

Afirmamos que

$$\langle G, s, t \rangle \in CAMINO \Longleftrightarrow \langle \varphi \rangle \in \overline{2SAT}\,.$$

En efecto, si existe un camino de $s = v_0, v_1, \ldots, v_l = t$ en $G$, y dado que $x_s$ y $x_t$ deben necesariamente tomar los valores $V$ y $F$ respectivamente para que $\varphi$ se pueda satisfacer, entonces alguna de las cláusulas $\overline{x_{v_i}} \vee x_{v_{i+1}}$ no se podrá satisfacer. Por otra parte, si no existe un camino entre $s$ y $t$ en $G$, entonces asignándole el valor $V$ a todas las variables asociadas a nodos en $G$ que se pueden alcanzar desde $s$ y $F$ a las restantes variables, obtenemos una asignación de valores de verdad que hace cierta a $\varphi$. Esto completa la demostración de la afirmación.

La característica altamente local de la construcción de $\varphi$ dados $G$, $s$ y $t$, permiten concluir que la reducción es a espacio logarítmico.

(iii.1).- Por Savitch, NL $\subseteq$ ESPACIO$(\log^2 n)$. Por el Teorema de la Jerarquía del Espacio, sabemos que ESPACIO$(\log^2 n)$ esta estrictamente contenido en ESPACIO$(\log^3 n) \subseteq$ *poli*L. Sigue que NL esta estrictamente contenido en *poli*L, en particular, NL $\neq$ *poli*L.

(iii.2).- Por contradicción, supongamos que P $=$ *poli*L. Sea $B$ un lenguaje P–completo bajo reducciones a espacio logarítmico. Como P $=$ *poli*L se tendrá que $B \in$ ESPACIO$(\log^j n)$ para algún $j \geq 1$. Además, si $A \in$ P, entonces $A \leq_L B$. El mismo argumento usado para probar que la reducción a espacio logarítmico es transitiva, permite concluir que si $A \leq_L B$ y $B \in$ ESPACIO$(\log^j n)$, entonces $A \in$ ESPACIO$(\log^j n)$. En otras palabras, *poli*L $\subseteq$ ESPACIO$(\log^j n)$, lo que contradice el Teorema de la Jerarquía del Espacio.

(iv).- El mismo argumento usado para probar que RP $\subseteq$ NP permite concluir que RPESPACIO $\in$ NPESPACIO $=$ PESPACIO. Como además es evidente que PESPACIO $\subseteq$ RPESPACIO, resulta que la clase RPESPACIO $=$ PESPACIO $=$ *co*RPESPACIO. Es decir, tanto RPESPACIO como *co*RPESPACIO corresponden trivialmente a clases conocidas.

PROBLEMA 2:

(i).- 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 $\omega$ 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 $\omega$, 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 $\omega$ 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 $1/2(1-2^{-p(|\omega|)}) + 1/2^{p(|\omega|)} > 1/2$.

(ii).- Como $\mathrm{ZPP} = \mathrm{RP} \cap co\mathrm{RP}$ y $\mathrm{RP} \subseteq \mathrm{NP}$, entonces $\mathrm{ZPP} \subseteq \mathrm{NP}$. Veamos que bajo las hipótesis del enunciado se tiene que $\mathrm{NP} \subseteq \mathrm{ZPP}$. En efecto, sea $A \in \mathrm{NP}$. Por hipótesis $A \leq_R L$. Luego, existe una mTND $M$ a tiempo polinomial $p$ con las características del enunciado pero donde $B$ se ha reemplazado por $L$. Por otro lado, existe una mTP $Z$ a tiempo esperado polinomial $q$ que decide el lenguaje $L$. Consideremos entonces la máquina $M'$ que en la entrada $\omega$ realiza lo siguiente:

1. Se repite el siguiente paso hasta que se genera una salida $\sigma$:

   > Simula $M$ en la entrada $\omega$ 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 $M$.

2. Simula $Z$ en la entrada $\sigma$, acepta si $Z$ acepta, y rechaza si $Z$ rechaza.

Observar que cada una de las repeticiones en el paso (1) toma tiempo $p(|\omega|)$. El número esperado de repeticiones de dicho paso hasta que se genere una salida corresponde a la esperanza de una variable aleatoria de parámetro $1/2$ (la fracción de las ramas de $M$ en las que se genera una salida). Luego, el tiempo esperado que $M'$ ocupa en el paso (1) es $2p(|\omega|)$. Notar además que dado que $M$ es a tiempo $p$, en caso de generarse una salida $\sigma$, esta tendrá un largo a lo más $p(|\omega|)$. El tiempo esperado que $M'$ ocupa en el paso (2) es por lo tanto $q(|\sigma|) = p(q(|\omega|))$, es decir, polinomial en $|\omega|$. Por último, es fácil ver que $M'$ acepta $\omega$ si y sólo si $Z$ acepta $\sigma$. Pero esto ocurre sólo si $\omega \in A$. Resumiendo, hemos establecido que $A \in \mathrm{ZPP}$.

(iii).- Como RP es cerrado bajo reducciones a tiempo polinomial, bastará mostrar que *SAT* está en RP. Asumiendo que $\mathrm{NP} \subseteq \mathrm{BPP}$ y como $SAT \in \mathrm{NP}$, sigue que existe una mT $M$ a tiempo polinomial tal que

$$\begin{aligned} \langle\varphi\rangle \in SAT \quad &\Longrightarrow \quad \mathbb{P}_\rho\left(M(\omega,\rho) = ac\right) \geq 1 - \frac{1}{3|\langle\varphi\rangle|}\,, \\ \langle\varphi\rangle \notin SAT \quad &\Longrightarrow \quad \mathbb{P}_\rho\left(M(\omega,\rho) = ac\right) \leq \frac{1}{3|\langle\varphi\rangle|}\,. \end{aligned}$$

Consideremos entonces la máquina probabilista $M'$ que recibe una entrada $\langle\varphi\rangle$, donde $\varphi = \varphi(x_1,\ldots,x_n)$ es una fórmula Booleana en las variables $x_1,\ldots,x_n$. La máquina $M'$ tratará de construir una asignación $a_1,\ldots,a_n$ para las variables $x_1,\ldots,x_n$ de la fórmula $\varphi$. La máquina $M'$ aceptará si y sólo si tiene éxito en la construcción de tal asignación. De esta manera se tendrá que si $\langle\varphi\rangle \notin SAT$, entonces $M'$ aceptará con probabilidad 0. El punto más delicado es garantizar que si $\langle\varphi\rangle \in SAT$, la máquina $M'$ pueda construir la asignación salvo por un evento cuya probabilidad es pequeña. Específicamente, $M'$ procede de la siguiente manera en la entrada $\langle\varphi\rangle$:

1. Determina el número de variables $n$ de $\varphi$.
2. Para $i = 1, \ldots, n$.
   - Simula $M$ en $\langle \varphi(x_1, \ldots, x_n) \rangle$ donde $x_j = a_j$ para $1 \leq j < i$, y $x_i = 1$.
   - Si $M$ acepta en el paso anterior, entonces le asigna a $a_i$ el valor 1 y de lo contrario le asigna el valor 0.
3. Acepta si $\varphi(a_1, \ldots, a_n) = 1$, en caso contrario rechaza.

Claramente, $M'$ acepta con probabilidad 0 toda palabra $\langle \varphi \rangle \notin SAT$. Por otro lado, si $M'$ no acepta $\langle \varphi \rangle \in SAT$, necesariamente debe ocurrir que para algún $i$,

$$\langle \varphi(a_1, \ldots, a_{i-1}, x_i, \ldots, x_n) \rangle \in SAT\,, \quad \text{y} \quad \langle \varphi(a_1, \ldots, a_i, x_{i+1}, \ldots, x_n) \rangle \notin SAT\,. \tag{1}$$

Lo anterior ocurre sólo si $M$ se equivoca en la entrada $\langle \varphi(a_1, \ldots, a_{i-1}, 1, \ldots, x_n) \rangle$, lo que ocurre con probabilidad a lo más $1/3|\langle \varphi \rangle| \leq 1/n$. La probabilidad que para algún $i \in \{1, \ldots, n\}$ ocurra una situación como la indicada en (1) es a lo más $n(1/3n) = 1/3$. Luego, $M'$ acepta con probabilidad al menos $2/3$ toda palabra $\langle \varphi \rangle \in SAT$. Resumiendo, $SAT \in$ RP.