Ma5201 Complejidad Computacional **14 de Julio de 2012**

Pauta Examen

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

PROBLEMA 1:

(i).- Para ver que NorCVAL está en P basta recordar que la evaluación de un circuito se puede hacer eficientemente (en tiempo polinomial) en el modelo RAM.

Sabemos que MonCVAL es P-completo.[1] Para concluir que NorCVAL es P-duro, bastará establecer que MonCVAL $\leq_{\mathrm{L}}$ NorCVAL. Sea $\langle C, \omega \rangle$ instancia de MonCVAL. Primero construimos un circuito C' reemplazando cada puerta lógica de C del tipo $\star \in \{\vee, \wedge\}$ con k arcos incidentes por un árbol como el que se muestra en la Figura 1. Posteriormente, construimos C'' a partir de C' reemplazando cada puerta lógica del tipo $\vee$ (respectivamente $\wedge$) por 2 (respectivamente 3) puertas lógicas del tipo NOR de acuerdo a lo que se muestra en la Figura 2). Notar que $C''(x) = C(x)$ cualquiera sea $x \in \{0,1\}^n$, en particular $C''(\omega) = C(\omega)$. Además, el tamaño de C' es a lo más cuadrático en el tamaño de C', y el tamaño de C'' es a lo más 3 veces el de C'. Luego, C'' es polinomial en el tamaño de C. Más aún, dado que las construcciones de C' y C'', a partir de C y C' respectivamente, son altamente locales, estas pueden ser fácilmente implementadas en espacio logarítmico. Sigue que existe un transformación a espacio logarítmico de $\langle C, \omega \rangle$ a $\langle C'', \omega \rangle$ tal que $C(\omega) = C''(\omega)$. En particular, $\langle C, \omega \rangle \in$ MonCVAL si y sólo si $\langle C'', \omega \rangle \in$ NorCVAL. Resumiendo, MonCVAL $\leq_{\mathrm{L}}$ NorCVAL como se quería establecer.

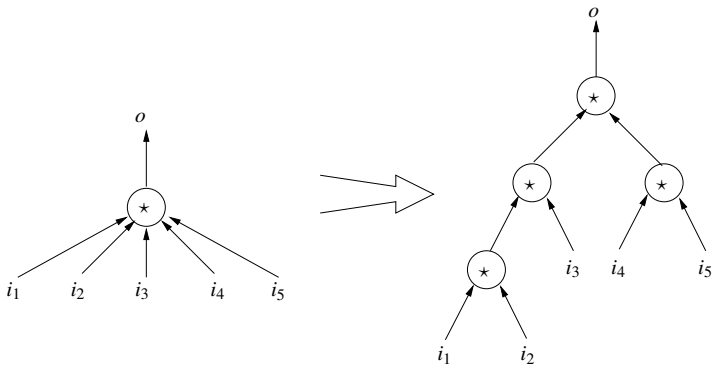

Figura 1: Reemplazo de una puerta lógica con múltiples arcos incidentes por un árbol binario de puertas lógicas del mismo tipo.

La reducción descrita de MonCVAL a NorCVAL no es la única posible, pero si la más simple de describir. Pequeñas adaptaciones a la reducción vista dan lugar a una reducción a espacio logarítmico de CVAL a NorCVAL, estableciendo así también que NorCVAL es P-duro.

[1] Recordar que MonCVAL es el lenguaje conformado por instancias del tipo $\langle C, \omega \rangle$ donde C es un circuito Booleano en n entradas con puertas lógicas del tipo $\wedge$ y $\vee$, $\omega \in \{0,1\}^n$, y $C(\omega) = 1$.

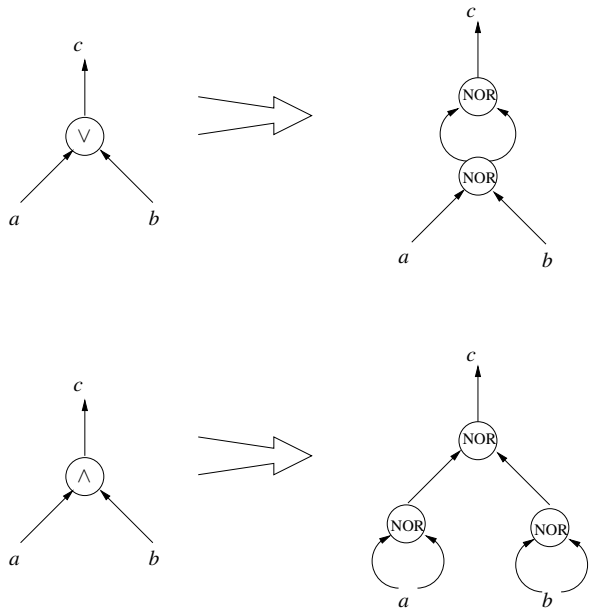

Figura 2: Reemplazo de puertas lógicas del tipo $\vee$ y $\wedge$ por puertas lógicas del tipo NOR.

(ii).- Veamos primero que RectTiling está en NP (lo cual, en este problema no es del todo inmediato). Sea $\langle (A,B);(a_1,b_1),\ldots,(a_k,b_k)\rangle$ una instancia de RectTiling. Consideremos el certificado $(x_1,y_1),\ldots,(x_k,y_k)$ tal que $0 \leq x_i \leq A$ y $0 \leq y_i \leq B$, $i \in \{1,\ldots,k\}$, donde (x_i,y_i) representa la posición de la esquina inferior izquierda del rectángulo (a_i,b_i) relativa a la esquina inferior izquierda del rectángulo principal (A,B). Afirmamos que si la instancia de RectTiling pertenece efectivamente al lenguaje, entonces debe existir una solución factible donde $x_1,\ldots,x_k,y_1,\ldots,y_k$ son todos enteros. En efecto, basta notar que si los pares (x_i,y_i)'s representan una dispoición válida de los rectángulos secundarios, entonces los $(\lfloor x_i \rfloor, \lfloor y_i \rfloor)$'s también son una disposición válida (verificar!). Sigue que existe un certificado de pertenencia de una instancia en RectTiling que es de tamaño a lo más $O(k(\log A + \log B))$, es decir polinomial en el largo de la instancia. La verificación del certificado requiere comprobar que:

- Para todo $i,j \in \{1,\ldots,k\}$, $i \neq j$ se tiene que

$$x_i \leq x_j \leq x_i + a_i \Longrightarrow ((y_i + b_i \leq y_j) \vee (y_j + b_j \leq y_i)) \,.$$

- Para todo $i \in \{1,\ldots,k\}$, se tiene que $x_i + a_i \leq A$ y que $y_i + b_i \leq B$.

Luego, la verificación requiere $O(k^2)$ operaciones artiméticas con números de largo acotado por $O(\text{máx}\{\log A, \log B\})$, por lo que puede ser fácilmente implementada en tiempo polinomial en el modelo RAM.

Veamos ahora que RectTiling es NP-duro. Bastará reducir SubsetSum a RectTiling. Sea $\langle \{s_1,\ldots,s_{k-1}\};t\rangle$ una instancia de SubsetSum. Dado que $\langle \{s_1,\ldots,s_{k-1}\};t\rangle \in$ SubsetSum si y sólo si $\langle \{s_1,\ldots,s_{k-1}\};\sum_{i=1}^{k-1} - t\rangle \in$ SubsetSum, sin pérdida de generalidad podemos asumir que $2t \leq \sum_{i=1}^{k-1} s_i$. Consideremos la instancia de RectTiling en que:

- $B = 2$.
- $A = \sum_{i=1}^{k-1} s_i - t$.
- $(a_i, b_i) = (s_i, 1)$ para todo $i \in \{1, \ldots, k-1\}$.
- $(a_k, b_k) = \left(\sum_{i=1}^{k-1} s_i - 2t, 1\right)$.

Notar que si $\langle \{s_1, \ldots, s_{k-1}\}; t \rangle \in$ SubsetSum y $2t \leq \sum_{i=1}^{k-1} s_i$, entonces existe un $I \subseteq \{1, \ldots, k-1\}$ tal que $\sum_{i \in I} s_i = t$. Sigue que el rectángulo principal puede ser cubierto colocando en una hilera los rectángulos secundarios $\{(a_i, b_i) : i \in I\} \cup \{(a_k, b_k)\}$ y en una segunda hilera el resto de los rectángulos secundarios. Luego, la reducción de más arriba lleva instancias en SubsetSum a instancias en RectTiling. El converso también se tiene. En efecto, sea $\langle (A, B); (a_1, b_1), \ldots, (a_k, b_k) \rangle$ una instancia en RectTiling que se obtiene de la reducción de una instancia de SubsetSum, digamos $\langle \{s_1, \ldots, s_{k-1}\}; t \rangle$. Observar que el área total de los rectángulos secundarios es

$$\sum_{i=1}^{k} a_i b_i = 2 \left(\sum_{i=1}^{k-1} s_i - t \right) = A \cdot B .$$

Luego, necesariamente los rectángulos (a_i, b_i)'s deben formar dos hileras, una que cubre el rectángulo principal hasta la altura 1 y otra que cubre el resto del rectángulo principal. Sean $(s_i, 1)$ con $i \in I \subseteq \{1, \ldots, k-1\}$ los rectángulos secundarios que están en la misma fila que el rectángulo secundario (a_k, b_k). Se tiene que,

$$\sum_{i \in I} a_i = A - a_k = \left(\sum_{i=1}^{k-1} s_i - t \right) - \left(\sum_{i=1}^{k-1} s_i - 2t \right) = t .$$

Por lo tanto, $\langle \{s_1, \ldots, s_{k-1}\}; t \rangle \in$ SubsetSum. Es fácil ver que la reducción descrita es a tiempo polinomial (de hecho, a espacio logarítmico) dado que el cálculo de la reducción sólo conlleva operaciones aritméticas simples y en su mayoría que dependen sólo de partes de la instancia sobre la que se está calculando la reducción.

(iii).- Sea $A = (A_{i,j})_{i,j}$ matríz de $n^2 \times n^2$ tal que los $A_{i,j}$'s pertenecen a $\{\square\} \cup [n^2]$. A continuación construiremos una reducción que le asocia a $\langle A \rangle$ la instancia $f(\langle A \rangle)$ de SAT. Antes de describir la reducción, nos será útil definir un fórmula Booleana, denotada φ, sobre n^2 variables Booleanas, digamos $y_1, \ldots, y_{n^2}$, tal que $\varphi(y_1, \ldots, y_{n^2}) = 1$ si y sólo si una y sólo una de las variables $y_1, \ldots, y_{n^2}$ toma el valor 1, equivalentemente,

$$\varphi(y_1, \ldots, y_{n^2}) \equiv \bigvee_{i \in [n^2]} \left(y_i \wedge \left(\wedge_{j \neq i} \overline{y_j} \right) \right) .$$

Sea entonces $f(\langle A \rangle) = \langle \varphi_A \rangle$ donde φ_A es una fórmula Booleana sobre n^6 variables $x_{i,j,k}$, con $i, j, k \in [n^2]$ y $x_{i,j,k} = 1$ representando que inicialmente $A_{i,j} = k$ o que $A_{i,j} = \square$ se reemplaza por k. Más aún,

$$\varphi_A \equiv \psi_A \wedge \left(\bigwedge_{i,j \in [n^2]} \varphi_{i,j}^{(cel)} \right) \wedge \left(\bigwedge_{i,k \in [n^2]} \varphi_{i,k}^{(row)} \right) \wedge \left(\bigwedge_{j,k \in [n^2]} \varphi_{j,k}^{(col)} \right) \wedge \left(\bigwedge_{s,t \in [n], k \in [n^2]} \varphi_{s,t,k}^{(blck)} \right) ,$$

donde ψ_A será verdadera sólo si $x_{i,j,k} = 1$ para todo $i, j, k \in [n^2]$ tal que $A_{i,j} = k$, $\varphi_{i,j}^{(cel)}$ será verdadera si y sólo si una y sólo una de las variables $x_{i,j,1}, \ldots, x_{i,j,n^2}$ es verdadera (es decir, $A_{i,j}$ toma un único valor en $[n^2]$), $\varphi_{i,k}^{(row)}$ será verdadera si y sólo si una y sólo una de las variables $x_{i,1,k}, \ldots, x_{i,n^2,k}$ es verdadera (es decir, en

la fila i de la matríz A un único $A_{i,j}$ toma el valor $k \in [n^2]$), $\varphi_{j,k}^{(col)}$ será verdadera si y sólo si una y sólo una de las variables $x_{1,j,k}, \ldots, x_{n^2,j,k}$ es verdadera (es decir, en la columna j de la matríz A un único $A_{i,j}$ toma el valor $k \in [n^2]$), y $\varphi_{s,t,k}^{(blck)}$ será verdadera si y sólo si una y sólo una de las variables $x_{i,j,k}$ con $(s-1)n < i \leq sn$ y $(t-1)n < j \leq tn$ es verdadera (es decir, en el bloque $B_{s,t}$ de la matríz A un único $(B_{s,t})_{i,j}$ toma el valor $k \in [n^2]$). Específicamente,

$$\begin{aligned}
\Psi_A &\equiv \bigwedge_{i,j \in [n^2]: A_{i,j} \neq \square} x_{i,j,A_{i,j}}\,, \\
\varphi_{i,j}^{(cel)} &\equiv \varphi\left(x_{i,j,k} : k \in [n^2]\right)\,, \\
\varphi_{i,k}^{(row)} &\equiv \varphi\left(x_{i,j,k} : j \in [n^2]\right)\,, \\
\varphi_{j,k}^{(col)} &\equiv \varphi\left(x_{i,j,k} : i \in [n^2]\right)\,, \\
\varphi_{s,t,k}^{(blck)} &\equiv \varphi\left(x_{i,j,k} : (s-1)n < i \leq sn, (t-1)n < j \leq tn\right)\,.
\end{aligned}$$

Es fácil ver que, dado A, cada una de las fórmulas Booleanas involucradas en la definición de φ_A son fáciles de construir en tiempo polinomial (de hecho, en log espacio), dado que las construcciones involucran esencialmente sólo ciclar sobre los índices adecuados. Claramente, hay una identificación uno a uno entre formas de completar una instancia de Sudoku dada por una matríz A con las características del enunciado y asignaciones de valores de verdad de $x_{i,j,k}$ con $i,j,k \in [n^2]$ que satisfacen φ_A. En resumen, f es una reducción a tiempo polinomial de Sudoku a SAT.

PROBLEMA 2:

(i).- Sea $A \in \mathbb{F}^{n \times n}$. Por definición de $\mathrm{Perm}(\cdot)$ y álgebra elemental se tiene que,

$$\mathrm{Perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)} = \sum_{i=1}^{n} a_{1,i} \sum_{\sigma \in S_n : \sigma(1)=i} \prod_{j=2}^{n} a_{j,\sigma(j)} = \sum_{i=1}^{n} a_{1,i} \sum_{\phi \in S_{n-1}} \prod_{j=1}^{n-1} (A_{1,i})_{j,\phi(j)} = \sum_{i=1}^{n} a_{1,i} \mathrm{Perm}(A_{1,i})\,.$$

(ii).- Sean $s,t \in \{1,\ldots,n-1\}$. Como $\mathbb{F}$ es un cuerpo de cardinalidad $p > n$ sabemos, de una consecuencia del Teorema Fundamental del Álgebra, que existe un polinomio $p \in \mathbb{F}[x]$ de grado a lo más n tal que $p_{s,t}(\mathbf{i}) = (A_{1,i})_{s,t}$ para todo $i \in \{1,\ldots,n\}$. La matríz $D_A(x) = (p_{s,t}(x))_{s,t}$ existe y satisface las condiciones del enunciado. Finalmente, observar que $\mathrm{Perm}(D_A(x))$ es una sumatoria de productos de $n-1$ términos del tipo $p_{s,t}(x)$. Luego, como cada producto es un polinomio de grado a lo más $n(n-1) < n^2$, sigue que $\mathrm{Perm}(D_A(x))$ es un polinomio de grado a lo más n^2.

(iii).- El probador honesto debe elegir $\mathrm{Perm}(D_{A_m}(x))$ como polinomio Q_m. Como A_m es un matríz de tamaño $(n-m+1) \times (n-m+1)$, de la parte anterior sabemos que dicho polinomio tendrá grado a lo más $(n-m+1)^2$ y que será tal que

$$\begin{aligned}
\sum_{i=1}^{n-m+1} (A_m)_{1,i} Q_m(\mathbf{i}) = \sum_{i=1}^{n-m+1} (A_m)_{1,i} \mathrm{Perm}(D_{A_m}(\mathbf{i})) &= \sum_{i=1}^{n-m+1} (A_m)_{1,i} \mathrm{Perm}((A_m)_{1,i}) \\
&= \mathrm{Perm}(A_m) = \mathrm{Perm}\left(D_{A_{m-1}}(\mathbf{b_{m-1}})\right) = Q_{m-1}(\mathbf{b_{m-1}}) = k_m\,.
\end{aligned}$$

Sigue entonces que si $\langle A,p,q,k\rangle$ pertenece a $L_{\mathbf{perm}}$ y el probador es honesto, entonces el verificador del sistema interactivo de demostración del enunciado acepta con probabilidad 1.

Por otro lado, si $\langle A, p, q, k\rangle$ no está en $L_{\mathbf{perm}}$, entonces para que el verificador del sistema interactivo del enunciado acepte debe ocurrir que para algún $m \in \{1, \ldots, n-1\}$ se tendrá que $Q_m(x) \neq \mathrm{Perm}\,(D_{A_m}(x))$ y $Q_m(\mathbf{b_m}) = \mathrm{Perm}\,(D_{A_m}(\mathbf{b_m}))$. Como $\mathbf{b_m}$ está elegido al azar en $\{1, \ldots, p\}$, y dado que dos polinomios distintos de grado a lo más $(n-m+1)^2$ pueden coincidir en a lo más $(n-m+1)^2$ valores de $\{\mathbf{1}, \ldots, \mathbf{p}\}$, se tendrá que

$$\mathbb{P}_{\mathbf{b_m}}\left(\, Q_m(\mathbf{b_m}) = \mathrm{Perm}\,(D_{A_m}(\mathbf{b_m}))\,|\, Q_m(x) \neq \mathrm{Perm}\,(D_{A_m}(x))\right) \leq \frac{(n-m+1)^2}{p}\,.$$

Por sub-aditividad de $\mathbb{P}\,(\cdot)$, sigue que la probabilidad que el verificador acepte está acotada por:

$$\sum_{m=1}^{n-1} \frac{(n-m+1)^2}{p} < \frac{1}{p} \sum_{m=1}^{n} m^2 < \frac{(n+1)^4}{4p}\,.$$

Tomando entonces $p = 3(n+1)^4/4$ se tiene que la última expresión está acotada por $1/3$ y que el protocolo del enunciado coloca a $L_{\mathbf{perm}}$ en IP.[2]

(ii).- Sea M una máquina tipo BPP que decide L en tiempo polinomial $p(n)$ y con error a lo más $1/2^{n+1}$ (dicha máquina existe por definición de BPP y aplicación del Lema de Amplificación). Decimos que $\rho \in \{0,1\}^{p(n)}$ es malo para $\omega \in \{0,1\}^n$ si $M(\omega, \rho) \neq L(\omega)$, donde $L(\omega) = 1$ si $\omega \in L$ y $L(\omega) = 0$ en caso contrario. Sigue que cualquiera que sea $\omega \in \{0,1\}^n$ hay a lo más $2^{p(n)}/2^{n+1}$ valores de $\rho \in \{0,1\}^{p(n)}$ malos para ω. Por lo tanto, hay a lo más $2^{p(n)-1}$ valores de ρ que son malos para algún $\omega \in \{0,1\}^n$. Luego, como $\{0,1\}^{p(n)}$ tiene cardinalidad $2^{p(n)}$, se tiene que existe $\rho_0 \in \{0,1\}^{p(n)}$ bueno para todo $\omega \in \{0,1\}^n$.

Como $M(\cdot,\cdot)$ es un máquina de Turing a tiempo polinomial, por el Teorema de Cook-Levin, sigue que existe una familia (uniforme) de circuitos Booleanos $(C_n(\cdot,\cdot))_{n\in\mathbb{N}}$ y un polinomio $q(\cdot)$ tales que $|C_n| \leq q(n)$ y para la cual $M(\omega,\rho) = C_n(\omega,\rho)$ cualquiera sea $\omega \in \{0,1\}^n$ y $\rho \in \{0,1\}^{p(n)}$. Definimos $C'_n(\cdot) = C_n(\cdot, \rho_0)$. Se tiene entonces que $(C'_n)_{n\in\mathbb{N}}$ es una familia de circuitors Booleanos en n entradas tal que $|C'_n| \leq q(n)$ para la cual $C'_n(\omega) = M(\omega, \rho_0) = L(\omega)$. Equivalentemente, $(C'_n)_{n\in\mathbb{N}}$ es una familia de circuitos Booleanos de tamaño polinomial que decide L, i.e. $L \in \mathrm{P/poli}$.

[2] En rigor, también se necesita argumentar que los pasos del protocolo del enunciado pueden sera realizados en tiempo polinomial por el verificador, pero no se pedía establecer esto último.