Ma50b Complejidad Computacional	**14 de Junio de 2008**

Pauta Control 3

Prof. Cátedra: M. Kiwi — *Prof. Auxiliar: P. Camacho*

PROBLEMA 1:

(i).- Primero veamos que **SubISO** está en NP. Para ello consideramos $\langle H, G\rangle$ donde $H = (U, F)$ y $G = (V, E)$ son grafos. Un *witness* de la pertenencia de $\langle H, G\rangle$ en **SubISO** lo constituye una función inyectiva φ de U en V representada como $\{(u, \varphi(u)) : u \in U\}$. La verificación consiste en comprobar que para todo $vw \in F$ se tiene que $\varphi(v)\varphi(w) \in E$. Se observa fácilmente que tanto el *witness* como el tiempo de verificación están polinomialmente relacionados con el tamaño de $\langle H, G\rangle$.

Veamos ahora que **CLIQUE** $\leq_m^P$ **SubISO**. Primero observar que $\langle G, k\rangle \in$ **CLIQUE** si y solo si $\langle K_k, G\rangle \in$ **SubISO**, donde K_k denota el grafo completo en k nodos. Luego, la reducción de **CLIQUE** a **SubISO** esta dada por la transformación que a $\langle G, k\rangle$, donde $0 \leq k \leq n$ con n el número de nodos de G, le asocia $\langle K_k, G\rangle$. Como la construcción de K_k puede realizarse en tiempo polinomial en $k \leq n$, sigue que la reducción es a tiempo polinomial en el tamaño de $\langle G, k\rangle$.

(ii).- Definiremos una transformación que a $\langle G\rangle$, donde $G = (V, E)$ es un grafo en n nodos, le asocia la fórmula Booleana φ_G en $2n$ variables $x_{v,1}, x_{v,2} \in \{0, 1\}$ para $v \in V$. Específicamente,

$$\varphi_G(x_{v,1}, x_{v,2} : v \in V) = \bigwedge_{uv \in E} (\neg(x_{v,1} \iff x_{u,1}) \vee \neg(x_{v,2} \iff x_{u,2})) .$$

Afirmamos que $\langle G\rangle \in$ **4COLOR** si y solo si $\langle\varphi_G\rangle \in$ **SAT**. En efecto, sea $\langle G\rangle$ tal que $G = (V, E)$ y $c : V \to \{0, 1, 2, 3\}$ tal que $uv \in E$ si y solo si $c(u) \neq c(v)$. Definimos $x_{v,1}x_{v,2}$ como la representación en binario de $c(v)$. Como $c(u) \neq c(v)$ es equivalente a $\neg(x_{v,1} \iff x_{u,1}) \vee \neg(x_{v,2} \iff x_{u,2})$, sigue que si $\langle G\rangle \in$ **4COLOR**, entonces $\langle\varphi_G\rangle \in$ **SAT**. Supongamos ahora que $\langle\varphi_G\rangle \in$ **SAT**. Sea $(x_{v,1}, x_{v,2} : v \in V)$ tal que $\varphi_G(x_{v,1}, x_{v,2} : v \in V) = 1$. Definimos $c : V \to \{0, 1, 2, 3\}$ tal que $c(v)$ corresponde al número cuya representación en binario es $x_{v,1}x_{v,2}$. Como para todo $uv \in E$ se tiene que $\neg(x_{v,1} \iff x_{u,1}) \vee \neg(x_{v,2} \iff x_{u,2}) = 1$, sigue que $c(u) \neq c(v)$ si $uv \in E$, i.e. $\langle G\rangle \in$ **4COLOR**.

De la discusión previa sigue que la transformación que a $\langle G\rangle$ le asocia $\langle\varphi_G\rangle$ es una reducción de **4COLOR** a **SAT**. La característica altamente local de la construcción de φ_G a partir de G permiten concluir que la transformación es a tiempo polinomial (de hecho, a espacio logarítmico).

(iii).- Supongamos que P = DESPACIO(n). Sea $L \in$ PESPACIO. Sigue que existe una máquina de Turing M a espacio polinomial, digamos $p(n)$, que decide L. Definimos

$$L' \quad = \quad \left\{\omega' : \omega' = \omega\#0^m, |\omega'| = p(|\omega|), \omega \in L\right\} .$$

Afirmamos que $L' \in$ DESPACIO(n). En efecto, sea M' la máquina de Turing que en la entrada ω' verifica que esta tiene la forma $\omega\#0^m$, calcula $p(|\omega|)$, verifica que $|\omega'| = p(|\omega|)$, simula M en ω y acepta si M acepta. La afirmación se obtiene observando que M' es a espacio lineal.

Resumiendo, tenemos que si $L \in$ PESPACIO, entonces $L' =$ DESPACIO$(n) =$ P. Afirmamos ahora que se debe tener que PESPACIO = P. En efecto, sea nuevamente $L \in$ PESPACIO decidido por una máquina de

Turing M a espacio $p(n)$, p polinomio. Sea M' la máquina de Turing a tiempo polinomial que decide L'. Sea M'' tal que en la entrada ω calcula $m = p(|\omega|) - |\omega| - 1$, simula M' en $\omega\#0^m$ y acepta si M' acepta. Se verifica que M'' decide L y es a tiempo polinomial en $|\omega|$. Luego, $L \in \mathrm{P}$ y se concluye entonces la afirmación enunciada.

Por el supuesto con que partimos y la discusión anterior se tiene que PESPACIO $=$ P $=$ DESPACIO(n).

PROBLEMA 2:

(i.1).- Si P $=$ NP, entonces existe un polinomio p tal que **SAT** es a tiempo $p(n)$. Sea φ una fórmula Booleana en n variables. Se verifica fácilmente que el Algoritmo 1 encuentra una asignación de valores de verdad que hace cierta a φ si tal asignación existe. Además, el algoritmo es a tiempo $O(n \cdot p(|\langle\varphi\rangle|))$ en la entrada $\langle\varphi\rangle$, i.e. es polinomial en el tamaño de su entrada.

Algorithm 1 Algoritmo para encontrar asignaciones de valores de verdad suponiendo que NP $=$ P.

procedure FINDSAT(φ) $\triangleright$ φ fórmula Booleana en las variables $x_1, \ldots, x_n$
  if $\langle\varphi\rangle \notin$ **SAT then**
    return NUL
  end if
  for $i \in \{1, \ldots, n\}$ **do**
    if $\langle\varphi(a_1, \ldots, a_{i-1}, 1, x_{i+1}, \ldots, x_n)\rangle \in$ **SAT then**
      $a_i \leftarrow 1$
    else
      $a_i \leftarrow 0$
    end if
  end for
  return $(a_1, \ldots, a_n)$
end procedure

(i.2).- Sea L el lenguaje de las palabras $\langle n, \alpha, \beta\rangle$ tales que $n, \alpha, \beta \in \mathbb{N}$ están codificados en binario y son tales que existe un $m \in \mathbb{N}$, $1 \leq \alpha \leq m \leq \beta$, m divisor de n. Se verifica fácilmente que $L \in$ NP (dado que un witness para $\langle n, \alpha, \beta\rangle \in L$ está dado por $\alpha \leq m \leq \beta$ y la verificación requiere solamente comprobar que $m|n$). Asumiendo que P $=$ NP, se tiene que existe un polinomio p tal que L es a tiempo $p(\cdot)$. Si $n \in \mathbb{N}$, se verifica que el Algoritmo 2 encuentra un factor no trivial de n si tal factor existe. Además, el algoritmo es a tiempo $O(\log n \cdot p(\log n))$, i.e. polinomial en el tamaño de la representación binaria de n.

(ii).- Como coNL $=$ NL bastará probar que $\overline{SAT}$ es NL–completo. Para ello, veremos que **PATH** 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

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

Afirmamos que $\langle G, s, t\rangle \in$ **PATH** si y solo si $\langle\varphi\rangle \in \overline{2SAT}$. En efecto, si existe un camino $s = v_0, v_1, \ldots, v_l = t$ en G que va de s a t, y dado que x_s y x_t deben necesariamente tomar los valores V y F respectivamente para que φ 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

Algorithm 2 Algoritmo para encontrar factores no-triviales suponiendo que NP = P.

```
procedure FINDFACTOR(n)                                   ▷ n ∈ ℕ, n > 2
    if ⟨n, 2, n − 1⟩ ∈ L then
        return NUL
    end if
    α ← 2
    β ← n − 1
    while α < β do
        γ ← ⌈(β − α)/2⌉
        if ⟨n, α, α + γ⟩ ∈ L then
            β ← α + γ
        else
            α ← α + γ
        end if
    end while
    return α
end procedure
```

verdad que hace cierta a φ. Esto completa la demostración de la afirmación.

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