

Ejercicio 1

Defino el siguiente lenguaje:

$$\text{EQUIV}(+, \circ) = \{(r_1, r_2) \mid r_1 \text{ y } r_2 \text{ son regex que solo utilizan los operadores unión y concatenación } (+ \text{ y } \circ) \text{ y } L(r_1) = L(r_2)\}$$

Demuestre que $\text{EQUIV}(+, \circ)$ es CONP -completo.

PASOS:

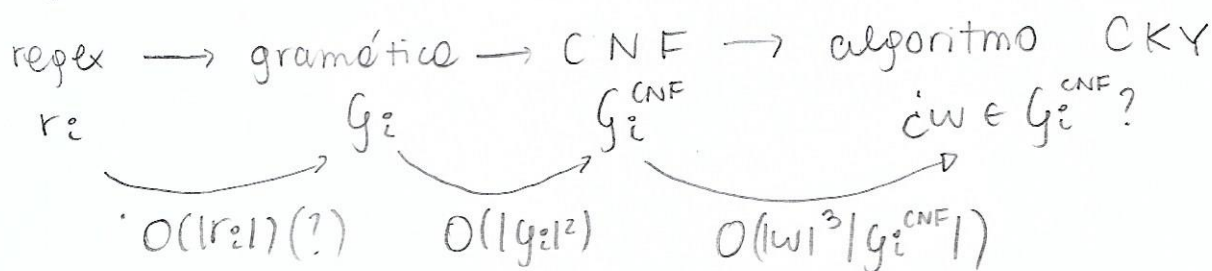
- ① Mostrar que está en CONP
- ② Mostrar que es CONP -completo (CoNP-hard).

① Sabemos que $L \in \text{CONP}$ ssi $\bar{L} \in \text{NP}$.

$$\overline{\text{EQUIV}(+, \circ)} = \{(r_1, r_2) \mid r_1 \text{ y } r_2 \text{ son regex que utilizan } + \text{ y } \circ, \text{ y } L(r_1) \neq L(r_2)\}$$

Certificado: $w \in L(r_1)$ tq $w \notin L(r_2)$ (SPG)

→ Verificación del certificado en $\text{poly}(|r_1|, |r_2|)$:



pero $|w| \leq |r_2|$, entonces esto es $O(\text{poly}(|r_2|))$.

Luego, $\overline{\text{EQUIV}(+, \circ)} \in \text{NP}$

$\Rightarrow \text{EQUIV}(+, \circ) \in \text{CoNP}$

② Por demostrar: $\text{EQUIV}(+, \cdot)$ es CONP -completo.

Obs: L es NP completo ssi $\bar{L}$ es CONP completo.

Dem. Sabemos que $L \in \text{NP}$ y $\forall A \in \text{NP}$, existe una función f computable en $O(n^k)$ tal que $\forall w \in \Sigma^*$

$$\begin{aligned} w \in A & \text{ ssi } f(w) \in L \\ \Leftrightarrow w \notin A & \text{ ssi } f(w) \notin L \\ \Leftrightarrow w \in \bar{A} & \text{ ssi } f(w) \in \bar{L} \end{aligned}$$

Como $\bar{\bar{A}} = A$, $\bar{\bar{L}} = L \in \text{NP}$, entonces $\bar{A}, \bar{L} \in \text{CoNP}$.

Luego, $\bar{L} \in \text{CoNP}$ y $\forall B (= \bar{A}) \in \text{CoNP}$, existe una función $g (= f)$ computable en $O(n^k)$ tq $\forall w \in \Sigma^*$

$$w \in B \text{ ssi } f(w) \in \bar{L}$$

Entonces $\bar{L}$ es CONP completo. El sentido contrario es análogo.

Reducción de $3\text{CNF-SAT} \rightarrow \text{EQUIV}(+, \cdot)$

Por obs. $\left(\begin{array}{l} \rightarrow \text{Como } 3\text{CNF-SAT es NP-completo, entonces } \overline{\text{EQUIV}(+, \cdot)} \\ \text{será NP completo (ya es NP por ①). y por la obs.} \\ \text{EQUIV}(+, \cdot) \text{ será CONP-completo.} \end{array} \right)$

Sea φ una fórmula en lógica proposicional de n variables proposicionales $\{p_1, \dots, p_n\}$ en 3CNF ($\varphi = \bigwedge_{i=1}^k C_i$, $k \in \mathbb{N}$)

Preguntas clave:

- ¿Cuándo φ es satisfacible en estas condiciones?
- ¿Qué puede "definir" una cláusula?

Idea de la reducción: Por cada cláusula C_i , hacer una regex R_i que "define (negativamente)" a C_i , es decir, $L(R_i)$ es el lenguaje que representa todas las valuaciones σ tales que $\sigma(C_i) = 0$.

Supuestos:

- φ es 3CNF de n variables, en donde todas las cláusulas están tienen los literales ordenados (de cualquier forma):

$$\checkmark (X_1 \vee X_2 \vee X_3) \wedge (X_2 \vee X_3 \vee X_4)$$

$$\times (X_1 \vee X_2 \vee X_3) \wedge (X_2 \vee X_1 \vee X_4)$$

- Las variables no se repiten en distintos literales de la misma cláusula: $\times (X_1 \vee \neg X_1 \vee X_2)$

→ Revisemos en $O(n^3)$ como en DNF y descartamos cláusulas.

- $C_i \neq C_j \quad \forall i \neq j$

→ Revisemos igual que antes.

- Decimos que $l \in C_i = (l_{i1} \vee l_{i2} \vee l_{i3})$ si $l = l_{i1} \vee l = l_{i2} \vee l = l_{i3}$

Función de reducción

- Por cada cláusula C_i escribimos $R_i = r_{i1} \cdot r_{i2} \cdot \dots \cdot r_{in}$, donde

$$r_{ij} = \begin{cases} 1, & \text{si } \neg p_j \in C_i \\ 0, & \text{si } p_j \in C_i \\ (0+1), & \text{eoc.} \end{cases}$$

- Si $\varphi = \bigwedge_{i=1}^K C_i$ entonces escribimos: $R_{\varphi} = R_1 + \dots + R_K$.
 $\varphi \rightarrow R_{\varphi}$ es computable en $O(\text{poly}(|\varphi|))$.

Ejemplo: $\bar{p} = \{p_1, p_2, p_3, p_4\}$, $\varphi = (\neg p_1 \vee p_2 \vee \neg p_3) \wedge (p_1 \vee p_2 \vee p_4)$

$$\rightarrow R_1 = 101(0+1)$$

$$R_2 = 00(0+1)0$$

$$\rightarrow R = R_1 + R_2 = 101(0+1) + 00(0+1)0$$

Analizando esta construcción, podemos demostrar que si σ es valoración y $w_\sigma = \sigma(p_1) \cdots \sigma(p_m)$, entonces para cualquier cláusula C_i de φ

LEMA

$$w_\sigma \in L(R_i) \text{ si } \sigma(C_i) = 0.$$

Dem

($\Rightarrow$) Sea $w_\sigma \in L(R_i)$. Podemos suponer que $R_i = (0+1)^{j-1} \alpha(p_j) (0+1)^{k-j} \alpha(p_k) (0+1)^{m-k} \alpha(p_m) (0+1)^{n-m}$

en donde

por construcción

$$C_i = l_{ij} \vee l_{ik} \vee l_{im}$$

literal
con
 p_j

literal
con
 p_k

literal
con
 p_m

$$0 \leq j < k < m \leq n$$

$$\alpha(p_s) = \begin{cases} 1, & \text{si } l_i = \neg p_s \\ 0, & \text{si } l_i = p_s \end{cases}$$

$$, s = j, k, m$$

$$\text{luego, } \sigma(p_j) = \alpha(p_j)$$

$$\sigma(p_k) = \alpha(p_k)$$

$$\sigma(p_m) = \alpha(p_m)$$

(d es cómo se construyó)

$$\left. \begin{array}{l} \sigma(p_j) = \alpha(p_j) \\ \sigma(p_k) = \alpha(p_k) \\ \sigma(p_m) = \alpha(p_m) \end{array} \right\} \text{ y } \sigma(C_i) = 0 \vee 0 \vee 0 = 0$$

($\Leftarrow$) Sea σ tq $\sigma(C_i) = \sigma(l_{ij} \vee l_{ik} \vee l_{im}) = 0$

luego,

$$\sigma(p_s) = \begin{cases} 0, & \text{si } l_{is} = p_s \\ 1, & \text{si } l_{is} = \neg p_s \end{cases}$$

$$, s = j, k, m$$

y $w_\sigma \in L(R_i)$, ya que los otros valores están libres. $\neq$

Proposición: φ es satisfacible si $L(R) \neq L((0+1)^n)$
 $(\Leftrightarrow) \varphi \in \text{3CNF-SAT}$ si $f(\varphi) \in \overline{\text{EQUIV}(+, \circ)}$
 , con $f(\varphi) = (R, (0+1)^n)$

Dem

$(\Rightarrow)$

Por contrapositivo, demostraremos que si $L((0+1)^n) = L(R)$
 $\Rightarrow \varphi$ no es satisfacible.

Sea σ una valoración cualquiera, y sea $w_\sigma = \sigma(p_1) \cdots \sigma(p_n)$.

Como $w_\sigma \in L((0+1)^n)$

$\Rightarrow w_\sigma \in L(R) = L(R_1 + \cdots + R_k)$, k cláusulas

En particular, $w_\sigma \in L(R_i)$ para algún $1 \leq i \leq k$.

Por el Lema visto anteriormente, necesariamente

$$\sigma(C_i) = 0$$

$$\Rightarrow \sigma(\varphi) = \sigma(C_1 \wedge \cdots \wedge C_k) = 0$$

Como σ es cualquier valoración sobre $\bar{p} = \{p_1, \dots, p_n\}$, entonces

φ es una contradicción y por ende no satisfacible. $\parallel (\Rightarrow)$

$(\Leftarrow)$

Supongamos que $L((0+1)^n) \neq L(R)$.

$\Rightarrow \exists w = a_1 \cdots a_n \in L((0+1)^n)$ tal que $w \notin L(R)$

Luego, para todo $1 \leq i \leq k$ (k cláusulas) $w \notin L(R_i)$

$\Rightarrow$ por el Lema anterior, $\sigma(C_i) = 1 \quad (\forall i)$

Si tomamos $\sigma(p_j) = a_j$, $j = 1, \dots, n$, se cumple que $\sigma(\varphi) = 1$. Luego, φ es satisfacible.

$\parallel (\Leftarrow)$



Luego, hemos demostrado que $\overline{\text{EQUIV}(+,\circ)} \in \text{NP}$ y además NP-hard. Luego, es NP-completo.

Por la observación inicial,

$\Rightarrow \overline{\text{EQUIV}(+,\circ)} = \text{EQUIV}(+,\circ)$ es coNP-completo.

