

Complejidad Computacional

Tarea 2

Karla Adriana Esquivel Ramírez
Andrea Itzel González Vargas
Luis Pablo Mayo Vega
Carlos Gerardo Acosta Hernández

Entrega: 14/03/17
Facultad de Ciencias UNAM

Ejercicios

1.

2.

Partamos de la afirmación de que \mathbf{P} es cerrado bajo complemento.

Dem. ($\mathbf{NP} \subseteq \mathbf{coNP}$)

Sea $L \in \mathbf{NP}$. Dado que $\mathbf{P} = \mathbf{NP}$,

$\Rightarrow L \in \mathbf{P}$,

$\Rightarrow \tilde{L} \in \mathbf{P}$,

$\Rightarrow \tilde{L} \in \mathbf{NP}$ (por hipótesis).

Por la definición de \mathbf{coNP} , $L \in \mathbf{coNP}$.

Dem. ($\mathbf{coNP} \subseteq \mathbf{NP}$)

Sea $L \in \mathbf{coNP}$,

$\Rightarrow \tilde{L} \in \mathbf{NP}$,

$\Rightarrow \tilde{L} \in \mathbf{P}$ (por hipótesis),

$\Rightarrow L \in \mathbf{P}$, pues \mathbf{P} es cerrado bajo complemento.

Por el supuesto de que $\mathbf{P} = \mathbf{NP}$, $L \in \mathbf{NP}$.

Ahora que hemos probado ambas contenciones, ello demuestra la igualdad de los conjuntos bajo la condición de que $\mathbf{P} = \mathbf{NP}$. Es decir, si $\mathbf{P} = \mathbf{NP}$, entonces $\mathbf{NP} = \mathbf{coNP}$.

3.

Demuestra que el lenguaje

$SPACETM = \{ \langle M \rangle \langle \alpha \rangle 1^n \mid M \text{ es una MT que acepta } \alpha \text{ en espacio } n \}$
es PSPACE-completo.

P.D.

I) $SPACETM \in PSPACE$.

II) Cualquier $L \in PSPACE$ se puede reducir a SPACETM

Entrada $\langle M, \alpha, 1^n \rangle$

Construimos $M' \in TM$ para Simular M , M' con dos contadores.

I) Un contador para contar el espacio utilizado por M , el otro contador contará el número de pasos que va a ejecutar M . Cada vez después de que M' Simule un paso de M , actualiza los contadores y comprueba si el espacio utilizado es mayor que n o si las etapas que se ejecutan en M son mayores que 2^{cn} para alguna constante c que está relacionada con M .

Si cualquiera de las anteriores ocurre entonces se rechaza inmediatamente. Observemos que el número total de configuraciones posibles de M es como máximo 2^{cn} , por lo tanto, si M no se detiene en este número de pasos, entonces debe haber entrado en un bucle infinito y M' Rechaza, después de simular M corriendo en w , si M acepta w en espacio n , M' Acepta, en otro caso M' rechaza. Por definición de SPACETM, M' decide a SPACE. Para simular M solamente utilizamos espacio $O(n)$. Entonces $SPACETM \in PSPACE$

II) Ahora Observemos que para cualquier $L \in PSPACE$ donde L es un lenguaje existe una Máquina de Turing Determinista M que decide L usando espacio $s(n)$ el cual es una entrada polinomial de tamaño n . Consideremos la función $f: \{0,1\}^n \Rightarrow \{0,1\}^m$ tal que $f(x) = \langle M, x, 1^{s(n)} \rangle$. Donde m es polinomial de n . La función f puede ser computada en espacio y tiempo polinomial. Por definición de SPACETM $x \in L$ si y sólo si $f(x) \in SPACETM$. Así que SPACETM es PSPACE-hard entonces $SPACETM \in PSPACE - \text{completo}$

\therefore Es PSPACE-Completo.

...

4.

Demuestra $2SAT \in NL$

Primero veamos que para todo F en 2-SAT podemos construir una grafica dirigida G de la siguiente manera:

$V(G)$ son las literales de F y hay una arista de a a b en $E(G)$ si y solo si $(a' \vee b)$ esta en F

Con esta construccion veamos que:

F es satisfacible si y solo si G no tiene un ciclo donde exista una literal y su negacion.

\Rightarrow)

veamos que $a' \vee b$ es equivalente a $a \rightarrow b$

Supongamos que tenemos un ciclo que contiene a b y b' en G . Si esto pasa entonces por transitividad significa que $b \dots b'$ y que $b' \dots b$ por lo que tendríamos $b \wedge b'$ lo cual es una contradiccion.

Por lo que si F es satisfacible entonces G no tiene un ciclo donde exista una literal y su negacion.

\Leftarrow)

Veamos estos haciend induccion sobre el numero de variables Cuando se tienen 0 variables

Si F no tiene variables entonces F es satisfacible y G no tiene ciclos.

Hip. Ind

Supongamos que se cumple cuando se tiene n variables

Ahora vemos que pasa cuando se tiene un variables mas:

Tomamos una literal b que en la grafica G generada no tiene un camino de b a b' .

Ahora supongamos que hay un camino de b a c y un camino de b a c'

Por contraposicion tendríamos un camino de c' a b' y por transitividad tendríamos un camino b a b' lo cual sera una contradiccion.

Ahora hagamos todas las literales a las que llegamos a partir de b verdaderas, esta conjunto esta bien definido por lo anterior visto. Entonces todas las clausulas que contenga a un literal de este conjunto se satisfacen. Y nos queda una grafica G' dada por las clausulas restantes las cuales tampoco tienen un ciclo con una literal y su negacion y por Hipotesis de induccion esta formula tambien es satisfacible.

\therefore Si G no tiene ciclos donde existe una literal y su negacion F es satisfacible

En libro tenemos que $NL = coNL$ veamos que 2-SAT' en NL implica 2-SAT en NL

Para ver si algo cumple con 2-SAT' basta con ver si hay una camino de alguna literal b a b' y de b' a b esto es que exista un ciclo en la grafica G generada. Por lo que podemos reducir el problema a PATH que de igual manera por lo visto en el libro PATH esta en NL .

\therefore 2-SAT' esta en $NL \therefore$ 2-SAT esta en NL

5.

6.

Supongamos que n es la cota superior de movimientos después de los cuales se tiene un ganador en el juego. Para que uno de los jugadores, digamos el jugador 1, tenga una estrategia ganadora en un juego de información perfecta -es decir, que los jugadores conocen todos los movimientos desde que comenzó el juego-, es necesario que para toda primera jugada del jugador 2, exista una serie de jugadas siguientes por parte del jugador 1 que al termino del juego le permitan ganar a dicho jugador.

Lo anterior lo podemos modelar considerando una Fórmula Booleana Cuantificada (QBF). Primero, consideremos que cada movimiento puede ser codificado como una cadena binaria, decidir

qué jugador gana al término del juego podemos expresarlo como una fórmula Booleana φ . S.p.g $\varphi = 1$, indica que 2 ha ganado el juego. Dicho esto, el jugador 2 tendrá una estrategia ganadora si y sólo si, la siguiente QBF es verdadera (aquí empieza el jugador 1):

$$QBF_1 := \forall x_1 \exists x_2 \forall x_3 \exists x_4 \dots \exists x_{2n} \varphi(x_1, x_2, \dots, x_n) \quad (1)$$

Supongamos que QBF_1 es falsa, con lo que su negación debe ser verdadera, por la definición de una QBF (al no tener variables libres), con lo que tendríamos:

$$QBF_2 := \neg(\forall x_1 \exists x_2 \forall x_3 \exists x_4 \dots \exists x_{2n} \varphi(x_1, x_2, \dots, x_n)) \quad (2)$$

Es decir:

$$QBF_2 := \exists x_1 \forall x_2 \exists x_3 \forall x_4 \dots \forall x_{2n} \neg \varphi(x_1, x_2, \dots, x_n) \quad (3)$$

Teníamos previamente que el jugador 2 ganaba, por lo que $\neg \varphi = 0$, así que el ganador en esta QBF_2 es 1 y se muestra que existe una estrategia en la que este jugador gana el juego. Por lo tanto, hemos demostrado que de los dos jugadores tiene una estrategia ganadora en el juego finito de información perfecta.