Pauta de Corrección

Segundo Certamen

Introducción a la Informática Teórica Informática Teórica

9 de julio de 2016

1. Una gramática sensible al contexto $G = (N, \Sigma, P, S$ tiene producciones de la forma:

   $$\alpha \to \beta$$

   donde α es una palabra cualquiera sobre $N \cup \Sigma$ que contiene al menos un no terminal, y $|\alpha| \leq |\beta|$. En resumen, aplicar una producción nunca acorta la forma sentencial.

   a) Podemos construir un LBA que aplique (nodeterministamente) las producciones "en reversa" partiendo de la palabra dada. Para cumplir con la descripción, al final de la cinta rellena con espacios que resulten de ir acortando la palabra. Si este proceso termina con únicamente el símbolo de partida en la cinta, acepta.

   b) Hay varias opciones. Una es simplemente generar sistemáticamente todas las formas sentenciales de la gramática partiendo del símbolo de partida. Al llegar a una forma sentencial de largo mayor que la palabra dada, sabemos que ese camino no puede llevar a generarla. Es claro que el conjunto de formas sentenciales posibles hasta un largo dado es finito, y por tanto puede revisarse exhausticamente. Esto corresponde a decidir el lenguaje, es recursivo.

Puntajes

Total			25
a) Esbozar LBA		15	
Producciones alargan la forma sentencial	5		
Aplicar producciones "en reversa"	10		
b) Esbozar decididor		10	
Formas sentenciales de interés son finitas	5		
Revisarlas todas da un decididor	5		

2. Llamemos $\mathsf{TM-RB}$ al problema de determinar si la TM rojo-azul M alguna vez prende la luz roja. Reducimos L_u a $\mathsf{TM-RB}$, como sabemos que L_u no es decidible, tampoco lo es $\mathsf{TM-RB}$.

   Dada una máquina de Turing M y una palabra ω, construimos una TM-RB M' que con cualquier entrada x entra a un estado rojo si y solo si M acepta ω. La construcción simplemente marca como rojo el estado final de M, y construye una TM-RB que descarta la entrada x, copia ω en la cinta y corre a M modificada sobre ésta. La figura 1 muestra la construcción resultante. Es claro que M'

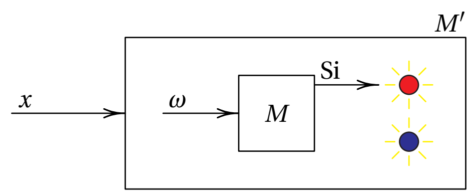

Figura 1: La construcción de M' dados M y ω

   prende la luz roja con alguna entrada (en realidad, cualquier entrada) si y solo si M acepta ω. La construcción esbozada es puramente mecánica y bastante simple, una máquina de Turing la puede llevar a cabo. O sea, $L_u \leq \mathsf{TM-RB}$, como prometimos demostrar.

Puntajes

Total		25
Reducir de un problema que se sabe no decidible	20	
Conclusión	5	

3. Debemos demostrar las tres propiedades enunciadas.

Reflexiva: Debemos demostrar que para todo problema A es $A \leq_p A$. Pero esto es trivial, la "reducción" consiste en no hacer nada, lo que claramente está acotado por un polinomio.

Transitiva: Debemos demostrar que para todos los problemas A, B, C si $A \leq_p B$ y $B \leq_p C$ entonces $A \leq_p C$.

Sean p_{AB}, p_{BC} los polinomios implícitos en las reducciones $A \leq_p B$ y $B \leq_p C$, respectivamente. Sea σ_A una instancia del problema A. Por la reducción $A \leq_p B$, en tiempo $p_{AB}(|\sigma_A|)$ construimos una instancia σ_B de B, donde sabemos que $|\sigma_B| \leq p_{AB}(|\sigma_A|)$.

A su vez, por la reducción $B \leq_p C$, en tiempo $p_{BC}(|\sigma_B|)$ construimos una instancia σ_C de C.

Combinando las anteriores, aplicando sucesivamente las reducciones $A \to B \to C$ traducimos σ_A en σ_C en tiempo acotado por $p_{BC}(p_{AB}(\sigma_A))$, que es un polinomio. O sea, $A \leq_p C$.

Antisimetría: Esta viene de regalo en la pregunta, por la definición de $A =_p B$.

Puntajes

Total		15
Reflexividad	3	
Transitividad	9	
Antisimetría	3	

4. Lo que haremos será armar un grafo en el cual los vértices que forman el conjunto independiente puedan tomar valor verdadero juntos, uno por cláusula (o sea, k es el número de cláusulas). Para ello conectamos entre sí los vértices que corresponden a una variable y su negación.

   El grafo resultante tiene un vértice por cada literal de la fórmula, y el número de arcos que un grafo puede tener está acotado por un polinomio en el número de vértices, el tamaño del grafo resultante es claramente polinomial en el tamaño de la fórmula original.

   Como ejemplo, tomemos la fórmula en 3CNF:

$$(x_1 \vee \overline{x}_2 \vee x_3) \wedge (x_2 \vee x_4 \vee \overline{x_5}) \wedge (\overline{x}_2 \vee x_3 \vee \overline{x}_4) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee x_5)$$

   Esta queda representada por el grafo de la figura 2. Es claro que si la expresión

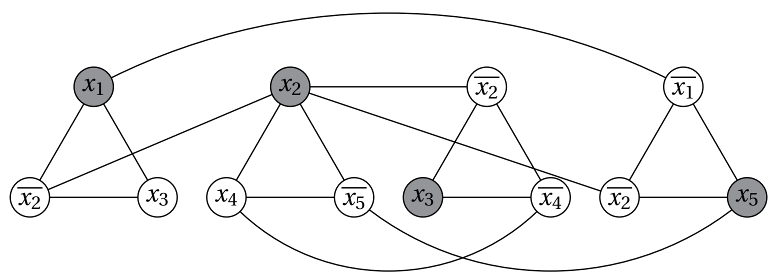

Figura 2: Grafo para una fórmula en 3CNF

   es satisfacible en el grafo resultante hay exactamente un vértice del conjunto independiente en cada cláusula, y para cada variable a lo más uno de x y $\overline{x}$ es verdadero (están conectados). Los vértices que son parte del conjunto independiente están marcados en gris en la figura 2, correspondiente a la asignación de valores $x_1 = V, x_2 = V, x_3 = V x_4 = F, x_5 = V$. O sea, k es el número de cláusulas.

Puntajes

Total		25
Explicación del diseño	10	
Grafo resultante es polinomial en el largo de la fórmula	5	
Satisfacible si y solo si el grafo tiene un conjunto independiente	10	

5. Cada punto por turno.

   a) Si $L \in \mathsf{P}$, quiere decir que hay un algoritmo determinista que en tiempo polinomial en $|\omega|$ determina si $\omega \in L$. Pero entonces determina también en tiempo polinomial si $\omega \notin L$, vale decir, $\overline{L} \in \mathsf{coNP}$.

   b) Considerando la pista, una manera de definir problemas en NPes dar un certificado para la instancia $\omega \in L$ de largo acotado por un polinomio en $|\omega|$ con el cual se puede verificar por un algoritmo determinista en tiempo polinomial si $\omega \in L$. De forma similar, para problemas en coNPdebemos tener un certificado de similares caracaterísticas de que $\omega \notin L$.

   Para FACTOR como certificado basta exhibir un factor d de x, $d \leq y$. Para verificar, basta dividir.

   Para $\overline{\mathsf{FACTOR}}$ un certificado es la factorización en primos de x. Sabemos que si escribimos x por ejemplo en decimal, la suma de los largos de los factores primos será cercana al largo de x, y sabemos que PRIME está en P, por lo que podemos verificar que cada factor es realmente primo, verificar que multiplicándolos obtenemos x, y que ninguno de ellos es menor a y en tiempo polinomial determinista.

   Como $\mathsf{FACTOR} \in \mathsf{NP}$ y $\overline{\mathsf{FACTOR}} \in \mathsf{NP}$ (o sea, $\mathsf{FACTOR} \in \mathsf{coNP}$), concluimos que $\mathsf{FACTOR} \in \mathsf{NP} \cap \mathsf{coNP}$.

Puntajes

Total			35
a) Discusión		10	
b) Demostración		25	
$\mathsf{FACTOR} \in \mathsf{NP}$	10		
$\overline{\mathsf{FACTOR}} \in \mathsf{NP}$	10		
Conclusión	5		