Pauta de Corrección

Certamen Recuperativo
Introducción a la Informática Teórica

30 de septiembre de 2015

1. Por turno.

   a) Un lenguaje es *recursivo* si es aceptado por una máquina de Turing determinista que siempre se detiene. Un lenguaje es *recursivamente enumerable* si es aceptado por una máquina de Turing.

   b) Un problema se llama NP-duro si todo problema en NP puede reducirse polinomialmente a él.

   c) Una *reducción* de un problema P_1 (básicamente la pregunta si $\sigma \in P_1$) al problema P_2 es un algoritmo (en rigor, máquina de Turing que siempre se detiene) que traduce σ a σ' tal que $\sigma \in P_1 \iff \sigma' \in P_2$.

   d) Un problema es no decidible si el lenguaje del caso no es recursivo. O sea, no hay una máquina de Turing que siempre se detiene que acepte el lenguaje del caso.

Puntajes

Total		20
a) Explicación/definiciones	6	
b) Definición	4	
c) Definición	5	
d) Definición	5	

2. Los lenguajes regulares son recursivos. La figura 1 esboza un reconocedor para la intersección entre $L_1 = \mathscr{L}(M_1)$

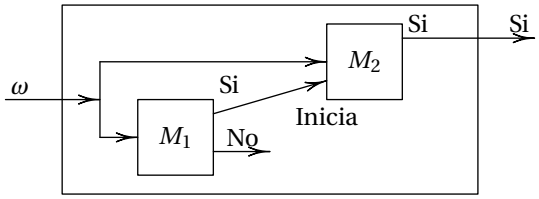

Figura 1: Intersección entre regular y recursivamente enumerable

para M_1 un DFA y $L_2 = \mathscr{L}(M_2)$ con M_2 una máquina de Turing. Esto demuestra lo solicitado.

Puntajes

Total	20
Lenguajes regulares son recursivos	5
Esbozo del reconocedor	10

3. Cada punto es esencialmente independiente de los demás.

   a) Vimos construcciones para "correr" partes de la cinta, que no requieren espacio extra. Al hacerlo, podemos "rellenar" al final con espacios. Con esto, dada una gramática sensible al contexto podemos construir un LBA que haga lo siguiente:

   - Nodeterministamente halle el lado derecho de una producción en la cinta.
   - Reemplaze el lado derecho por el izquierdo de esa producción. Como la gramática es sensible al contexto, el lado izquierdo no es más largo que el derecho, y podemos hacerlo sin requerir espacio adicional.
   - Si la cinta contiene solo el símbolo de partida, acepte.

   Esto reconoce el lenguaje generado por la gramática.

   b) Supongamos $L_1 = \mathscr{L}(M_1)$ y $L_2 = \mathscr{L}(M_2)$, con M_1 y M_2 LBAs. Siguiendo la pista, construimos un LBA con una cinta de dos pistas que actúa como sigue:

   - Copia el contenido de la entrada en la segunda pista, luego vuelve al comienzo.
   - Simula el funcionamiento de M_1 en la primera pista.
   - Si M_1 acepta, retrocede al comienzo de la cinta y simula el funcionamiento de M_2 en la segunda pista.
   - Si M_2 acepta, acepta.

   Esta construcción acepta $L_1 \cap L_2$, con lo que los lenguajes aceptados por LBA son cerrados respecto de intersección.

   c) No. El punto 3*a* muestra que los lenguajes sensibles al contexto son aceptados por LBA, pero pueden haber lenguajes aceptados por LBA que no son sensibles al contexto.

Puntajes

Total		30
a) Explicación/esbozo de construcción	10	
b) Explicación/esbozo de construcción	10	
c) Explicación	10	

4. Sabemos que la intersección entre lenguajes de contexto libre y lenguajes regulares es de contexto libre. Podemos construir un PDA que acepta $L_= = \{\#a = \#b\}$ (por ejemplo partiendo de la gramática dada en la pregunta), y con él y el DFA M dado podemos construir un PDA M_I que acepta $\mathscr{L}(M) \cap L_=$. Dado M_I podemos construir una gramática de contexto libre G_I, y determinar si $L_I = \varnothing$ (básicamente, determinando si su símbolo de partida es útil). La respuesta a la pregunta original es si $L_I \neq \varnothing$ (este es un ejemplo de reducción).

Puntajes

Total	20
Construcción y argumento que son algoritmos	20

5. Para demostrar que SET COVER es NP-completo, debemos demostrar que está en NP y que es NP-duro. Para demostrar que es NP-duro debemos demostrar que todos los problemas en NP pueden reducirse polinomialmente a él. Para ello basta reducir polinomialmente a SET COVER un problema NP-duro; como VERTEX COVER es NP-complejo es NP-duro, y basta demostrar VERTEX COVER $\leq_p$ SET COVER.

   Primero, SET COVER está en NP. Basta "adivinar" la colección de conjuntos y calcular su unión, verificando que es $\mathscr{U}$. El tiempo para hacer esto es a lo más k veces el tamaño de $\mathscr{U}$, que claramente es polinomial en el tamaño del problema.

   Para demostrar que es NP-duro, mostramos una reducción de VERTEX COVER a SET COVER. Consideremos un grafo $G = (V, E)$ y un entero k. Un conjunto $C \subseteq V$ es un VERTEX COVER de G si todos los vértices de G aparecen en los arcos que incluyen a C. Podemos tomar $\mathscr{U} = V$, los subconjuntos $\mathscr{S}_v = \{(u, v) : (u, v) \in E\}$, y buscamos $C \subseteq V$ tal que $|C| \leq k$ y

   $$\bigcup_{v \in C} \mathscr{S}_v = V$$

   Esta traducción es simple de hacer, requiere trabajo esencialmente lineal en el tamaño de G.

   Como está en NP y es NP-duro, es NP-completo.

Puntajes

Total	35
Discusión definiendo NP-completo	5
El problema está en NP	10
Reducción de VERTEX COVER a SET COVER	15
Conclusiones	5