Pauta de Corrección

Certamen Recuperativo
Introducción a la Informática Teórica

7 de septiembre de 2013

1. Cada cual por turno.

   a) Un lenguaje es *recursivo* si es acpetado por una máquina de Turing que siempre se detiene.

   b) Un *autómata finito no determinista* (NFA) es:

   $$M = (Q, \Sigma, \delta, q_0, F)$$

   donde:

   ***Q*:** Conjunto finito de *estados*

   Σ: *Alfabeto* de entrada

   $\boldsymbol{\delta : Q \times (\Sigma \cup \{\epsilon\}) \to 2^Q}$**:** *Función de transición*

   $\boldsymbol{q_0 \in Q}$**:** *Estado inicial*

   $\boldsymbol{F \subseteq Q}$**:** Conjunto de *estados finales*

   La idea es que M está en un estado q en cada instante, y en cada paso consume nada o un símbolo de la entrada (llamemos x a lo que consume), y pasa a alguno de los estados en $\delta(q, x)$. Inicialmente parte al comienzo del string a procesar, y si hay manera de llegar a consumir el string completo terminando en un estado en F, M lo acepta. En caso contrario lo rechaza.

   c) Un *problema no decidible* es un problema para el cual no existe un algoritmo (modelado por una máquina de Turing que siempre se detiene) que lo resuelva.

   d) Una *gramática* consta de:

   $$G = (N\Sigma, P, S)$$

   donde:

   ***N*:** Alfabeto de símbolos *no terminales*

   Σ: Alfabeto de símbolos *terminales*, $N \cap \Sigma = \varnothing$

   ***P*:** Conjunto finito de *producciones* de la forma $\alpha \to \beta$, donde $\alpha \in (N \cup \Sigma)^* N (N \cup \Sigma)^*$ (o sea, está formado por terminales y no terminales, con al menos uno de estos últimos), $\beta \in (N \cup \Sigma)^*$ (o sea, está formado por terminales y no terminales)

S $\in$ ***N*:** *Símbolo de partida*

La idea es que inicialmente tenemos únicamente el string S, luego en un *paso de producción* reemplazamos algún lado izquierdo de una producción por el correspondiente lado derecho. Si el resultado está formado únicamente por terminales, es un string del *lenguaje generado por la gramática.*

e) En una gramática se dice que un símbolo es *inútil* si nunca participa en la derivación de uno de los strings generados.

Puntajes

Total		20
Cada parte 5 puntos	20	

2. La jeraquía de Chomsky clasifica gramáticas (ver también $1d$) en cuatro niveles según la forma de sus producciones. En lo que sigue son $\alpha \to \beta$, con $\alpha \in (N \cup \Sigma)^* N (N \cup \Sigma)^*$ y $\beta \in (N \cup \Sigma)^*$. Las restricciones que se indican, de haberlas, se aplican a todas las producciones de la gramática.

   Tipo 0 (Irrestrictas): No hay restricciones.

   Tipo 1 (Sensibles al contexto): $|\alpha| \le |\beta|$.

   El nombre viene de que una forma alternativa de describirlas es mediante producciones de la forma $\alpha A \beta \to \alpha \gamma \beta$, donde $A \in N$, $\alpha, \beta \in (N \cup \Sigma)^*$ mientras $\gamma \in (N \cup \Sigma)^+$. Vale decir, A puede reemplazarse por γ en el contexto α y β.

   Tipo 2 (Contexto libre): Las producciones son todas de la forma $A \to \beta$, con $A \in N$ y $\beta \in (N \cup \Sigma)^+$.

   Tipo 3 (Regulares): Las producciones son de una de las formas $A \to \beta B$, con $A, B \in N$ y $\beta \in \Sigma^*$, o $A \to \beta$, con $\beta \in \Sigma^+$.

   Los autómatas respectivos son máquinas de Turing, autómatas linealmente acotados, autómatas de stack y autómatas finitos.

Puntajes

Total				20
Definición de gramática			5	
Tipo 0	Irrestrictas		4	
	Gramáticas	2		
	Autómata es TM	2		
Tipo 1	Sensibles al contexto		3	
	Gramáticas	3		
Tipo 2	Contexto libre		4	
	Gramáticas	2		
	Autómata es PDA no determinista	2		
Tipo 3	Regulares		4	
	Gramáticas	2		
	Autómata es NFA/DFA	2		

3. La demostración es por contradicción. Se simplifica si usamos el resultado de la pregunta 4. Si $\mathcal{L}$ fuera de contexto libre, lo sería $\mathcal{L}' = \mathcal{L} \cap a^+ b^+ c^+ d^+ = \{a^n b^n c^n d^n : n \geq 1\}$. Supongamos que $\mathcal{L}'$ es de contexto libre. Entonces es aplicable el lema de bombeo para CFL: Hay una constante N tal que si $\sigma \in \mathcal{L}'$ con $|\sigma| \geq N$ podemos escribir $\sigma = vwxyz$ con $0 < |wy| \leq N$ tales que $uw^k xy^k z \in \mathcal{L}'$ para todo $k \geq 0$.

   Elegimos $\sigma = a^N b^N c^N d^N \in \mathcal{L}'$, con $|\sigma| = 4N > N$, y supongamos una división como la indicada. Ahora bien, si w o y están formados por más de un tipo de símbolo, al repetirlo el string resultante no pertenece a $a^+ b^+ c^+ d^+$. Repitiendo w e y sólo podemos aumentar el número de símbolos de uno o dos tipos, no de los cuatro. Por tanto $uw^2 xy^2 z \notin \mathcal{L}'$, una contradicción.

Puntajes

Total		20
Demostración por contradicción	2	
Plantear lema de bombeo	5	
Elegir σ, cumple condiciones	5	
Suponer división, elegir k	5	
Llegar a contradicción	3	

4. Podemos suponer dado un DFA M_1 que acepta $\mathcal{L}_1$, o sea $\mathcal{L}_1 = \mathscr{L}(M_1)$. Igualmente podemos suponer dado un PDA M_2 que acepta $\mathcal{L}_2$ por estado final, $\mathcal{L}_2 = \mathscr{L}(M_2)$. Sean:

$$M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$$
$$M_2 = (Q_2, \Sigma, \Gamma, \delta_2, q_2, F_2)$$

Construimos un PDA que acepta $\mathcal{L}_1 \cap \mathcal{L}$ por estado final como sigue:

$$M_I = (Q_1 \times Q_2, \Sigma, \Gamma, \delta_I, (q_1, q_2), F_1 \times F_2)$$

Los estados de M_I son pares de estados, M_I simula la ejecución de M_1 y M_2 en paralelo. la idea es seguir la pista del estado de cada autómata en el componente respectivo del estado. Para todas las combinaciones de $p' \in Q_1$, $p'' \in Q_2$, $x \in \Sigma \cup \{\epsilon\}$ y $A \in \Gamma$ definimos:

$$\delta_I((p', p''), x, A) = \{((\delta_1(p', x), q''), \alpha) \colon (q'', \alpha) \in \delta_2(p'', \epsilon, A)\}$$

Es claro que M_I acepta si y sólo si M_1 y M_2 aceptan.

Puntajes

Total		25
DFA para $\mathcal{L}_1$	5	
PDA para $\mathcal{L}_2$	5	
PDA para $\mathcal{L}_1 \cap \mathcal{L}_1$	10	
Justificar que acepta $\mathcal{L}_1 \cap \mathcal{L}_1$	5	

5. Dado el LBA

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

una *descripción instantánea de M* es un string:

$$\eta \in \Gamma^* Q \Gamma^*$$

Por la definición de M, si parte procesando $\sigma \in \Sigma^*$, el largo de las descripciones instantáneas será siempre $|\sigma| + 1$, el conjunto de descripciones instantáneas de M procesando σ es un conjunto finito. Es posible enumerar exhaustivamente las posiciones posibles de M y verificar así si acepta o no. O sea, $\mathsf{A}_{\mathrm{LBA}}$ es decidible.

Puntajes

Total		25
Descripciones instantáneas del LBA son un conjunto finito	10	
Enumerar exhaustivamente las posiciones	10	
Como siempre responde en plazo finito, es decidible	5	

6. Claramente DenseSubgraph está en NP: Dado un conjunto de n vértices de $G = (V, E)$ representado mediante listas de adyacencia puede verificarse en tiempo proporcional a $|V|^2$ cuántos arcos hay entre ellos.

   Reducir Clique a DenseSubgraph es muy simple: Ver si hay una clique de n vértices es equivalente a determinar si hay un subgrafo de n vértices y $k = n(n-1)/2$ arcos. Calcular k claramente toma tiempo acotado por un polinomio en el tamaño de los datos. Como Clique es NP-completo, DenseSubgraph es BP-duro.

   Hemos demostrado que DenseSubgraph está en NP y que es NP-duro, así que es NP-completo.

Puntajes

Total		20
Está en NP	5	
Es NP-duro	10	
Concluir que es NP-completo	5	