Pauta de Corrección

Primer Certamen

Introducción a la Informática Teórica
Informática Teórica

19 de enero de 2016

1. Por turno.

   a) Es simple ver que las gramáticas siguientes dan lo solicitado:

   Para L_1:

   $$\begin{aligned} S &\to aSc \mid aAc \\ A &\to Ab \mid b \end{aligned}$$

   Vemos que $S \Rightarrow^* a^{m-1}Sc^{m-1} \Rightarrow a^m Ac^m \Rightarrow^* a^m b^n c^m$, realmente no hay caminos alternativos.

   Para L_2:

   $$\begin{aligned} S &\to Sc \mid Ac \\ A &\to aAb \mid ab \end{aligned}$$

   Tenemos $S \Rightarrow^* Sc^{n-1} \Rightarrow Ac^n \Rightarrow^* a^m b^m c^n$, y nuevamente no hay caminos alternativos.

   b) Resulta $L_1 \cap L_2 = \{a^n b^n c^n : n \in \mathbb{N}\}$, que vimos en clase no es de contexto libre.

   Como si L_1 y L_2 son de contexto libre no siempre es de contexto libre $L_1 \cap L_2$, los lenguajes de contexto libre no son cerrados respecto de intersección.

Puntajes

Total			25
a) Gramáticas		16	
L_1 y explicación	8		
L_2 y explicación	8		
b) $L_1 \cap L_2$		9	
$L_1 \cap L_2 = \{a^n b^n c^n : n \in \mathbb{N}\}$	6		
$L_1 \cap L_2$ no es CFL, conclusión	3		

2. Supongamos que el lenguaje dado es de contexto libre, por lo que cumple el lema de bombeo. Sea N la constante del lema, elijamos una palabra σ del lenguaje:

   $$\sigma = a^N b^N a^N b^N$$

   Es $|\sigma| = 4N \geq N$, Por el lema de bombeo, hay u, v, w, x, y con $|vwx| \leq N$ y $vx \neq \epsilon$ tales que:

   $$\sigma = uvwxy$$

   y $uv^k wx^k y$ pertenece al lenguaje para todo $k \in \mathbb{N}_0$. Como $|vwx| \leq N$, al repetir v e x a lo más se pueden repetir las a y b de dos de las cuatro secuencias que forman σ, y el resultado no pertenece al lenguaje. Hemos llegado a una contradicción, el lenguaje no es de contexto libre.

Puntajes

Total		20
Aplicar lema de bombeo	5	
Elección de σ	5	
Condición sobre v, w, x	2	
Llegar a contradicción	8	

3. Por la pista, basta generar palabras de la forma:

$$\alpha x \alpha' \beta y \beta'$$

con $|\alpha| = |\alpha'|$ y $|\beta| = |\beta'|$ pero $x \neq y$. La siguiente gramática logra lo pedido:

$$\begin{aligned} S &\to A \mid B \\ A &\to aaA \mid abA \mid baA \mid bbA \mid a \mid b \\ B &\to CD \mid DC \\ C &\to aCa \mid aCb \mid bCa \mid bCb \mid a \\ D &\to aDa \mid aDb \mid bDa \mid bDb \mid b \end{aligned}$$

Vemos que A genera todas las palabras de largo impar, mientras B genera palabras de largo par, pero cuyas mitades difieren a lo menos en un símbolo.

Como hallamos una gramática de contexto libre para $\overline{L}_3$, este lenguaje es de contexto libre. Pero su complemento L_3 no es de contexto libre por la pregunta 3. Concluimos que los lenguajes de contexto libre no son cerrados respecto de complemento.

Puntajes

Total		25
Plantear gramática correcta, según la pista	20	
Conclusión	5	

4. Consideramos lenguajes C, que es NP-completo; N, que está en NP; P que está en P; R que es recursivamente enumerable y D que es decidible (recursivo). Veamos cada caso:

   a) $\boldsymbol{R \le D}$**:**
   Esto nos dice que R es recursivo.

   b) $\boldsymbol{C \le_p N}$**:**
   Esto permite concluir que N es NP-completo

   c) $\boldsymbol{P \le_p N}$**:**
   No es novedad, P está en $\mathsf{P} \subseteq \mathsf{NP}$, y todo problema en NP puede reducirse polinomialmente a C por ser NP-completo.

   d) $\boldsymbol{\overline{R} \le C}$**:**
   Como la reducción indicada es arbitraria, solo es relevante que C es recursivo. Esta reducción permite concluir que $\overline{R}$ es recursivo. Pero siendo $\overline{R}$ recursivo, es recursivo R.

   e) $\boldsymbol{C \le P}$**:**
   No es novedad, la "reducción" puede hacer el trabajo pesado de resolver el problema, y resolver la instancia de P resultante en corto tiempo nada permite concluir sobre el tiempo necesario para resolver C.

Puntajes

Total		30
a) $R \le D$	6	
R es recursivo		
b) $C \le_p N$	8	
N es NP-duro y en NP		
N es NP-completo		
c) $P \le_p N$	4	
No es novedad		
d) $\overline{R} \le C$	8	
C es recursivo		
$\overline{R}$ es recursivo		
R es recursivo		
e) $C \le P$	4	
Reducción arbitraria hace el trabajo		

5. Cada punto en turno.

    a) Un lenguaje L está en P si podemos decidir en tiempo polinomial en el largo de σ si $\sigma \in L$. Pero entonces podemos decidir en tiempo polinomial si $\sigma \notin L$. Concluimos que si $L \in \mathsf{coP}$ entonces $L \in \mathsf{P}$, o sea $\mathsf{P} = \mathsf{coP}$.

    b) Sabemos que $\mathsf{P} \subseteq \mathsf{NP}$. Dado $L \in \mathsf{P}$, si podemos determinar en tiempo polinomial que $\sigma \in L$ podemos determinar en tiempo polinomial que $\sigma \in \overline{L}$, lo que es simplemente un caso particular de determinar en tiempo polinomial en una máquina no determinista. Vale decir, si $L \in \mathsf{P}$ entonces $L \in \mathsf{coNP}$. Uniendo las anteriores, tenemos $L \in \mathsf{NP}$ y $L \in \mathsf{coNP}$, con lo que $P \subseteq \mathsf{NP} \cap \mathsf{coNP}$.

    c) Determinar si la fórmula Φ está en $\mathsf{TAUTOLOGY}$ es lo mismo que determinar si $\overline{\Phi}$ es siempre falsa, que es exactamente el complemento de SAT. Como SAT es NP-completo, está en NP; concluimos que $\mathsf{TAUTOLOGY}$ está en coNP.

Puntajes

Total			30
a)		8	
Qué significa $L \in \mathsf{P}$	3		
Corresponde a $\overline{L} \in \mathsf{P}$	3		
Concluir $\mathsf{P} = \mathsf{coP}$	2		
b)		8	
Sabemos $\mathsf{P} \subseteq \mathsf{NP}$	2		
Demostrar $L \in \mathsf{P}$ implica $L \in \mathsf{coNP}$	3		
$P \subseteq \mathsf{NP}$ y $P \subseteq \mathsf{coNP}$ es $P \subseteq \mathsf{NP} \cap \mathsf{coNP}$	3		
c)		14	
$\Phi \in \mathsf{TAUTOLOGY}$ es equivalente a $\overline{\Phi} \in \overline{\mathsf{SAT}}$	5		
SAT es NP-completo, está en NP	5		
Conclusión por la reducción anterior	4		