Examen de Computabilidad y Complejidad
(CMC)
13 de junio de 2007

(I) CUESTIONES: (Justifique formalmente las respuestas)

1. ¿Es el lenguaje $\{x \mid (\exists y,z \in \{0,1\}^*)((\exists u \in \{0\}^*)((x = yuz) \wedge (|y| = |z| = |u|)\}$ incontextual?

   Estableceremos que este lenguaje no es incontextual demostrando que no cumple el Lema de Iteración. Supóngase que lo cumple y sea n la constante del Lema. Sea $z = 1^n0^n1^n$, esta palabra pertenece al lenguaje y $|z| \geq n$ tal y como exige el Lema. En cualquiera de las posibles factorizaciones de z, de acuerdo a las condiciones del Lema de Iteración, al iterar se obtiene una palabra de la forma $1^m0^k1^j$ donde $m = n \vee j = n$. En consecuencia al considerar la iteración 0 se obtiene una palabra más corta que z en la que o su longitud no es múltiplo de 3 o lo es pero, en la descomposición yuz con $|y| = |z| = |u|$, en u existen 1's bien provenientes de su parte derecha (caso en que $j = n$) o bien de su parte izquierda (caso en que $m = n$). En cualquier caso la palabra no pertenece al lenguaje y así este no es incontextual.

   (1.0 punto)

2. ¿Es el lenguaje $\{x \mid (\exists y \in \{0,1\}^*)((\exists z \in \{0\}^*)((x = yz) \wedge (|y| = |z|) \wedge (|y|$ *no es múltiplo de 3*$)\}$ incontextual?

   El subconjunto de los números naturales que no son múltiplos de 3 es $X = \{3n+1, 3n+2 \mid n \geq 0\}$, así $|y|$ *no es múltiplo de 3* se puede escribir como $|y| \in X$; además el que $|y| = |z|$ obliga a que por cada símbolo de y exista un símbolo en z. A partir de estas consideraciones vamos a proponer una gramática incontextual que genere al lenguaje dado por lo que quedará establecida su incontextualidad. La gramática propuesta es la siguiente

   $S \rightarrow AAASBBB \mid AB \mid AABB$
   $A \rightarrow 0 \mid 1$
   $B \rightarrow 0$

   Es inmediato ver que cualquier palabra del lenguaje puede generarse en esta gramática. Sea x una palabra del lenguaje, así x = yz en las condiciones del enunciado. Como $|y| \in X$ podemos factorizar $|y| = 3k + j$ con $j = 1$ o $j = 2$. Ahora podemos utilizar la producción $S \rightarrow AAASBBB$ k veces y seguidamente la producción $S \rightarrow AB$ si $j = 1$ o $S \rightarrow AABB$ si $j = 2$. Finalmente cada símbolo auxiliar A y B lo reescribiremos adecuadamente.

   $$S \Rightarrow^* A^{3k}SB^{3k} \Rightarrow A^{3k+j}B^{3k+j} \Rightarrow^* yz = x, \text{ donde } A^{3k+j} \Rightarrow^* y \wedge B^{3k+j} \Rightarrow^* z$$

   También es sencillo ver que cualquier palabra generada por la gramática pertenece al lenguaje. Sea $x \in \{0,1\}^*$ con $S \Rightarrow^* x$. Así, por la estructura de las producciones de la gramática, debe tenerse una derivación genérica de la forma $S \Rightarrow^* C^mSD^m \Rightarrow^* C^{m+j}D^{m+j} \Rightarrow^* x$, donde $C \in \{0,1,A\}$, $D \in \{0,B\}$, $m \in \{3n \mid n \geq 0\}$ y $j \in \{1,2\}$; por tanto, $m + j \in X$ y en consecuencia x pertenece al lenguaje ya que puede factorizarse como yz donde $C^{m+j} \Rightarrow^* y \wedge D^{m+j} \Rightarrow^* z$ con lo que $|y| = |z| \wedge |y| \in X$.

(1.0 punto)

3. Sea Σ un alfabeto y $L \subseteq \Sigma^*$. Sea R un lenguaje recursivo. Se define la operación $P(L) = \{x \in L \,/\, (x \notin R) \wedge (x = x^r)\}$.

   I. Si L es un lenguaje recursivamente enumerable ¿lo es también P(L)?

   II. Si L es un lenguaje recursivamente enumerable ¿lo es también L – P(L)?

   Sea el lenguaje $I = \{x \in \Sigma^* /\ x = x^r\}$. En primer lugar estableceremos que este lenguaje es recursivo; para este fin definiremos una máquina de Turing M_I que se detenga para cada entrada y que lo reconozca. La máquina M_I tien dos cintas y opera como sigue: cuando recibe una entrada x la recorre copiándola en la otra cinta, luego ubica el cabezal de una de las cintas al principio de la palabra (y lo desplaza de izquierda a derecha) y en la otra al final (y lo desplaza de derecha a izquierda) comenzando a comparar los símbolos correspondientes que encuentra cuando recorre las dos cintas. Si en este proceso encuentran los cabezales un blanco acepta, si en algún momento los símbolos leídos por los dos cabezales difieren rechaza.

   En lo sigue el complementario de un lenguaje L lo representaremos con L^c.

   I. Si L es un lenguaje recursivamente enumerable, entonces P(L) también lo es. Sea $W = R^c \cap I$, donde R^c, (por lo dicho) el complementario de R, es un lenguaje recursivo por ser el complemento una operación de cierre en la familia lenguajes recursivos; así W es también un lenguaje recursivo por ser la intersección otra operación de cierre. Basta ahora ver, inmediatamente a partir del enunciado, que $P(L) = L \cap W$. Puesto que W por ser recursivo es también recursivamente enumerable y ser la intersección una operación de cierre en los lenguajes recursivamente enumerables se sigue que P(L) es un lenguaje recursivemente enumerable.

   II. Si L es un lenguaje recursivamente enumerable, entonces también lo es L – P(L). Basta notar que, a partir de lo anterior, $L - P(L) = L - (L \cap W) = L \cap (L \cap W)^c = L \cap (L^c \cup W^c) = (L \cap L^c) \cup (L \cap W^c) = L \cap W^c$.

   (2.0 puntos)

4. Sean L_1, L_2, L_3 lenguajes. Se define la operación $\nabla(L_1, L_2, L_3)$ que define el lenguaje formado por las palabras que pertenecen al menos a dos de sus argumentos.

   I. ¿Es la operación ∇ una operación de cierre en la familia de los lenguajes recursivos?

   II. ¿Es la operación ∇ una operación de cierre en la familia de los lenguajes recursivamente enumerables?

   A partir de la definición pude verse que $\nabla(L_1, L_2, L_3) = (L_1 \cap L_2) \cup (L_1 \cap L_3) \cup (L_2 \cap L_3)$. Puesto que la intersección y la unión son operaciones de cierre tanto en los lenguajes recursivos como en los lenguajes recursivamente enumerables se tiene que la operación ∇ es de cierre en ambas familias de lenguajes.

   (1.0 punto)

(II) PROBLEMAS

5. Desarrolle un módulo *Mathematica*, adecuadamente explicado, que reciba como entrada una gramática incontextual y retorne True si todas sus producciones se

ajustan al formato $A \to BxC$ donde $B,C \in$ Auxiliares $\cup \{\lambda\}$ y $x \in$ Terminales*, retornando False en otro caso.

En principio debe notarse que todas las producciones $A \to \alpha$ con $|\alpha| \leq 2$ se ajustan al formato especificado de modo que la comprobación algorítmica explícita sólo deberá realizarse para aquellas producciones $A \to \alpha$ con $|\alpha| > 2$, lo que se traduce en comprobar, para cada uno de estos consecuentes, que excluidos el primer y el último símbolo el resto son todos terminales o, equivalentemente, no son auxiliares.

```
P5[G_List] := Module[{i, j, k},
    For[i = 1, i ≤ Length[G[[3]]], i++,
      For[j = 1, j ≤ Length[G[[3, i, 2]]], j++,
        For[k = 2, k < Length[G[[3, i, 2, j]]], k++,
          If[MemberQ[G[[1]], G[[3, i, 2, j, k]]],
             Return[False]]
        ]
      ]
    ];
    Return[True]
]
```

(2.0 puntos)

6. Dadas las gramáticas

   $G1 : S \to bSaS \mid c$

   $G2 : S \to aSb \mid aSaS \mid SSS \mid ba$

   y la sustitución $f(a) = L(G1) \cup L(G2)^r$, $f(b) = L(G2)^*$ y $f(c) = \{\lambda\}$, obténgase una gramática incontextual para el lenguaje $(f(L(G1)) - \{\lambda\})^+ f(L(G2))$.

$f(a)$: $S_a \to S_1 \mid S_2$

$S_1 \to bS_1aS_1 \mid c$

$S_2 \to bS_2a \mid S_2aS_2a \mid S_2S_2S_2 \mid ab$

$f(b)$: $S_b \to S_3S_b \mid \lambda$

$S_3 \to aS_3b \mid aS_3aS_3 \mid S_3S_3S_3 \mid ba$

$f(c)$: $S_c \to \lambda$

$f(L(G1))$: $S_4 \to S_bS_4S_aS_4 \mid S_c$

$f(L(G1)) - \{\lambda\}$: Símbolos anulables: S_4, S_b, S_c

$S_4 \to S_bS_4S_aS_4 \mid S_4S_aS_4 \mid S_bS_aS_4 \mid S_bS_4S_a \mid S_aS_4 \mid S_4S_a \mid S_bS_a \mid S_a$

$S_b \to S_3S_b \mid S_3$

$(f(L(G1)) - \{\lambda\})^+$: $S_5 \to S_4S_5 \mid S_4$

$f(L(G2))$: $S_6 \to S_aS_6S'_b \mid S_aS_6S_aS_6 \mid S_6S_6S_6 \mid S'_bS_a$

$S'_b \to S_3S'_b \mid \lambda$

$(f(L(G1)) - \{\lambda\})^+ f(L(G2))$:

$S_7 \to S_5S_6$

$S_5 \to S_4S_5 \mid S_4$

$S_4 \to S_bS_4S_aS_4 \mid S_4S_aS_4 \mid S_bS_aS_4 \mid S_bS_4S_a \mid S_aS_4 \mid S_4S_a \mid S_bS_a \mid S_a$

$S_b \to S_3S_b \mid S_3$

$S_3 \to aS_3b \mid aS_3aS_3 \mid S_3S_3S_3 \mid ba$

$S_a \to S_1 \mid S_2$

$S_1 \to bS_1aS_1 \mid c$

$S_2 \to bS_2a \mid S_2aS_2a \mid S_2S_2S_2 \mid ab$

$S_6 \to S_aS_6S'_b \mid S_aS_6S_aS_6 \mid S_6S_6S_6 \mid S'_bS_a$

$S'_b \to S_3S'_b \mid \lambda$

(1.5 puntos)

7. Dada la gramática G obtenga una gramática G' simplificada y en Forma Normal de Chomsky con L(G') = L(G) – {λ}.

$S \to AaAB \mid B \mid EE \mid BA \mid \lambda \mid a$	$A \to BS \mid CEC \mid ab \mid \lambda$
$B \to aa \mid AbB$	$C \to aC \mid D$
$D \to aC \mid C \mid CEE$	$E \to aC \mid CD \mid EE \mid \lambda$

Símbolos generativos: S, A, B, E

$S \to AaAB \mid B \mid EE \mid BA \mid \lambda \mid a$

$A \to BS \mid ab \mid \lambda$

$B \to aa \mid AbB$

$E \to EE \mid \lambda$

Todos los símbolos son alcanzables.

Símbolos anulables: S, A, E

Eliminación de las producciones λ

$S \to AaAB \mid aAB \mid AaB \mid aB \mid B \mid EE \mid E \mid BA \mid a$

$A \to BS \mid B \mid ab$

$B \to aa \mid AbB \mid bB$

$E \to EE \mid E$

Símbolos útiles: S, A, B

$S \to AaAB \mid aAB \mid AaB \mid aB \mid B \mid BA \mid a$

$A \to BS \mid B \mid ab$

$B \to aa \mid AbB \mid bB$

Eliminación de las producciones unitarias

$S \to AaAB \mid aAB \mid AaB \mid aB \mid aa \mid AbB \mid bB \mid BA \mid a$

$A \to BS \mid ab \mid aa \mid AbB \mid bB$

$B \to aa \mid AbB \mid bB$

Todos los símbolos son útiles.

Forma normal de Chomsky

$S \rightarrow AX_aAB \mid X_aAB \mid AX_aB \mid X_aB \mid X_aX_a \mid AX_bB \mid X_bB \;\; BA \mid a$

$A \rightarrow BS \mid X_aX_b \mid X_aX_a \mid AX_bB \mid X_bB$

$B \rightarrow X_aX_a \mid AX_bB \mid X_bB$

$X_a \rightarrow a$

$X_b \rightarrow b$

- - - - - - - - - - - - - -

$S \rightarrow AZ_1 \mid X_aZ_2 \mid AZ_3 \mid X_aB \mid X_aX_a \mid AZ_4 \mid X_bB \mid BA \mid a$

$A \rightarrow BS \mid X_aX_b \mid X_aX_a \mid AZ_4 \mid X_bB$

$B \rightarrow X_aX_a \mid AZ_4 \mid X_bB$

$X_a \rightarrow a$

$X_b \rightarrow b$

$Z_1 \rightarrow X_aZ_2$

$Z_2 \rightarrow AB$

$Z_3 \rightarrow X_aB$

$Z_4 \rightarrow X_bB$

(1.5 puntos)