Examen de Computabilidad y Complejidad
(CMC)
5 de febrero de 2009

(I) CUESTIONES: (Justifique formalmente las respuestas) [5.00 puntos]

1. ¿Es el lenguaje $\{a^n b^m c^k \ / \ (n = m) \vee (n < k)\}$ incontextual?

Nótese que el lenguaje L, a partir de su definición, puede inmediatamente descomponerese como $L = L_1 \cup L_2$, donde

$$L_1 = \{ a^n b^m c^k / n = m, k \geq 0\} \text{ y}$$
$$L_2 = \{a^n b^m c^k / n < k, m \geq 0\}.$$

Tanto L_1 como L_2 son lenguajes incontextuales ya que se pueden generar, por ejemplo, con las siguientes gramáticas incontextuales

- $L_1 = \{a^n b^n / n \geq 0\}\{c^k / k \geq 0\}$ puede así inmediatamente generarse con la gramática incontextual
  $$S_1 \rightarrow XY \qquad X \rightarrow a X b \mid \lambda \qquad Y \rightarrow cY \mid \lambda$$
- L_2 puede generarse también inmediatamente con la gramática incontextual
  $$S_2 \rightarrow a S_2 c \mid S_2 c \mid Zc \qquad Z \rightarrow bZ \mid \lambda$$

En consecuencia L es incontextual por ser la unión una operación de cierre en la familia de los lenguajes incontextuales.

(0.5 puntos)

2. Sean $L', L'' \subseteq \{a,b\}^*$, se define
   $$L = \{xcy \ / \ (x \in L') \wedge (y \in L'') \wedge (|x| = |y|)\}.$$
   I. Si L' y L'' son incontextuales ¿es L incontextual? *(1.5 puntos)*
   II. Si L' y L'' son recursivos ¿es L recursivo? *(0.75 puntos)*
   III. Si L' y L'' son recursivamente enumerables ¿es L recursivamente enumerable? *(0.75 puntos)*

$\boxed{\text{I}}$ Veamos que el lenguaje L no es, en general, incontextual. Para ello sean los lenguajes incontextuales $L' = \{a^n b^n / n \geq 0\}$ y $L'' = \{a^k / k \geq 0\}$, en este caso $L = \{ a^n b^n c a^{2n} / n \geq 0\}$. Podemos ahora establecer que el lenguaje L no es incontextual por medio de dos vías: 1) mediante el Lema de Iteración, y 2) reduciéndolo mediante operaciones de cierre a un lenguaje no incontextual. Lo haremos de la segunda forma reduciéndolo al lenguaje $\{a^n b^n c^n / n \geq 0\}$, que ya sabemos que no es incontextual. Sea el homomorfismo $h(a) = h(d) = a$, $h(b) = b$ y $h(c) = c$, sea $L_1 = h^{-1}(L) \cap a^*b^*cd^* = \{ a^n b^n c d^{2n} / n \geq 0\}$. Sea ahora el homorfismo $g(a) = a$, $g(b) = b$, $g(c) = \lambda$ y $g(d) = c$, se tiene que $g(L_1) = \{a^n b^n c^{2n} / n \geq 0\}$. Sea finalmente el homomorfismo $f(a) = a$, $f(b) = b$ y $f(c) = cc$, se tiene que $f^{-1}(L) = \{a^n b^n c^n / n \geq 0\}$.

$\boxed{\text{II}}$ Veamos que el lenguaje L es recursivo cuando L' y L'' lo son. Si L' y L'' son lenguajes recursivos, entonces existen, respectivamente, las máquinas de Turing M' y M'', que se detienen para cada entrada, y L(M') = L' y L(M'') = L''. Seguidamente definimos una máquina de Turing M, con alfabeto de entrada {a,b,c}, que se detendrá para cada entrada y L(M) = L, quedando así establecido que L es recursivo. La máquina M opera como sigue. Cuando se le proporciona una entrada z comprueba que en la misma existe una y sólo una aparición del símbolo c y en consecuencia que z puede factorizarse como $z = xcy$ con $x,y \in \{a,b\}^*$; seguidamente comprueba que |x| = |y|, para ello puede usar una cinta con un sector adicional sobre el que ir marcando secuencialmente la unificación de cada símbolo de x con el correpondiente símbolo de y. Si esta condición se satisface aplica el prefijo x a la máquina M', si ésta rechaza también se rechaza z, en otro caso se aplica el sufijo y a la máquina M", si M" acepta se acepta z, en otro caso se rechaza.

III Veamos que el lenguaje L es recursivamente enumerable cuando L' y L" lo son. El razonamiento es similar al del caso anterior, con las diferencias propias asociadas a los lenguajes recursivamente enumerables (las máquinas M' y M" ya no necesitan detenerse para cada entrada y la máquina M tampoco).

3. Dado un lenguaje `L` se define el lenguaje

$$\mu(L) = \{x \ / (x = x^r) \wedge (x \in \underline{\mathit{prefijos}}(L))\}.$$

Si `L` es recursivamente enumerable, entonces ¿lo es también $\mu(L)$?

Si el lenguaje L es recursivamente enumerable, entonces también lo es μ(L). Para demostralo describiremos una máquina de Turing multicinta M_L que lo reconozca. Puesto que L es recursivamente enumerable existe un generador de Turing G_L que lo genera y que se encuentra controlado durante la generación, en el sentido habitual, mediante una señal de control, de continuación, que le permite generar las palabras de una en una, con las correspondientes pausas, de acuerdo a esta señal. La máquina M_L opera como sigue. Cuando se le aplica una entrada `x` comprueba primeramente que $x = x^r$, para ello copia la palabra `x` en una cinta auxiliar y vuelve al principio de `x` en la cinta de entrada recorriendo seguidamente `x` y x^r síncronamente en sentidos contrarios, comprobando que los respectivos símbolos examinados son iguales; si en ambas cintas encuentra simultáneamente un blanco, y sólo en estas circunstancias, da por satisfecha la condición, esto es: $x = x^r$. Seguidamente, si esta condición se cumple, arranca el generador G_L que comienza la generación; siempre que genera una palabra se suspende hasta que recibe la señal de continuación; con la última palabra generada `z` comprueba que `x` es un prefijo de la misma, para ello escribe `z` en una cinta auxiliar y recorre desde el principio síncronamente `x` y `z` comprobando que los símbolos examinados coindicen; mientras esta coincidencia se dé continúa la comprobación que finaliza con la aceptación si se encuentra un blanco al examinar `x` (tanto si en `z` aparece un blanco como si no), en otro caso, si esta coincidencia de símbolos no se produce, se envía la señal de control al generador volviéndose a repetir esta fase.

(1.5 puntos)

(II) PROBLEMAS `[5.00 puntos]`

4. Desarrolle un módulo *Mathematica*, adecuadamente explicado, que reciba como entrada una gramática incontextual G y un símbolo auxiliar A de G y retorne `True` si se cumple que $A \Rightarrow_G^+ \alpha A \beta$, retornando `False` en otro caso.

El programa *Mathematica* pedido debe obtener y examinar el conjunto de símbolos que se pueden alcanzar en la gramática a partir de A en uno o más pasos; si entre los mismos encuentra A retorna `True`, en otro caso (puesto que este conjunto es finito cuando lo genere por completo) retorna `False`. De este modo podemos observar que se asemeja mucho al programa que se pedía en la segunda práctica de laboratorio para el cálculo de los símbolos alcanzables.

```
Math[G_List,A_Symbol] :=
  Module[{producciones,alcanzados,símbolos,nuevos,
          consecuentes,p,s,i,j},

    producciones = G[[3]];
    nuevos = alcanzados = {A};

    While[nuevos != {},
      símbolos = {};
      For[i = 1, i <= Length[nuevos], i++,
        p = Cases[producciones, {nuevos[[i]],_}];
        consecuentes = p[[1,2]];
        For[j = 1, j <= Length[consecuentes], j++,
          If[MemberQ[consecuentes[[j]],A], Return[True]];
          símbolos = Union[símbolos,
                      Intersection[consecuentes[[j]],G[[1]]]]
        ]
      ];
      nuevos = Complement[símbolos,alcanzados];
      alcanzados = Union[alcanzados,nuevos]
    ];

    Return[False]
  ]
```

(2.0 puntos)

5. Dadas las gramáticas

$G: S \to aSbS \mid aSb \mid c$ y $G': S \to aSS \mid SSb \mid \lambda$

y las sustituciones f y g definidas como sigue: $f(a) = L(G) \cup L(G)^r$, $f(b) = L(G)^*$; y $g(a) = f(a)$, $g(b) = f(b)$ y $g(c) = f(L(G'))$.

Obténgase una gramática incontextual para el lenguaje $L(G)g(L(G))$.

Aplicando estrictamente la teoría (sin aplicar ninguna simplificación evidente e inmediata) tenemos:

g(a): $A \to B \mid C$
$B \to aBbB \mid aBb \mid c$
$C \to CbCa \mid bCa \mid c$

g(b): $D \to ED \mid \lambda$
$E \to aEbE \mid aEb \mid c$

g(c):
f(a): $F \to H \mid K$
$H \to aHbH \mid aHb \mid c$
$K \to KbKa \mid bKa \mid c$

f(b): $L \to ML \mid \lambda$
$M \to aMbM \mid aMb \mid c$

$N \to FNN \mid NNL \mid \lambda$

$L(G)g(L(G))$: $S \to RT$
$R \to aRbR \mid aRb \mid c$
$T \to ATDT \mid ATD \mid N$

(1.5 puntos)

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

$S \to SAS \mid AD \mid AF \mid SS \mid BB$ $\quad$ $A \to AB \mid BC \mid CA \mid b$
$B \to BCS \mid BBF \mid \lambda$ $\quad$ $C \to a \mid abC \mid FC \mid B$
$D \to SFD \mid EAF \mid BBD$ $\quad$ $E \to EE \mid FE \mid a$
$F \to EF \mid DFE \mid D$

Aplicando estrictamente la teoría (sin aplicar ninguna simplificación evidente e inmediata) tenemos:

ELIMINACIÓN DE LOS SÍMBOLOS INÚTILES

Eliminación de los símbolos no generativos: D, F

$S \to SAS \mid SS \mid BB$ $\quad$ $A \to AB \mid BC \mid CA \mid b$
$B \to BCS \mid \lambda$ $\quad$ $C \to a \mid abC \mid B$
$E \to EE \mid a$

Eliminación de los símbolos no alcanzables: E

$S \to SAS \mid SS \mid BB$ $\quad$ $A \to AB \mid BC \mid CA \mid b$
$B \to BCS \mid \lambda$ $\quad$ $C \to a \mid abC \mid B$

ELIMINACIÓN DE LAS PRODUCCIONES-λ

Símbolos anulables: S, A, B ,C

$S \to SAS \mid AS \mid SA \mid SS \mid A \mid BB \mid B \mid S$
$A \to AB \mid B \mid BC \mid C \mid CA \mid b \mid A$
$B \to BCS \mid CS \mid BS \mid BC \mid S \mid C \mid B$
$C \to a \mid abC \mid ab \mid B$

ELIMINACIÓN DE LAS PRODUCCIONES UNITARIAS

$S \to SAS \mid AS \mid SA \mid SS \mid AB \mid BC \mid CA \mid b \mid BB \mid BCS \mid CS \mid BS \mid a \mid abC \mid ab$
$A \to SAS \mid AS \mid SA \mid SS \mid AB \mid BC \mid CA \mid b \mid BB \mid BCS \mid CS \mid BS \mid a \mid abC \mid ab$
$B \to SAS \mid AS \mid SA \mid SS \mid AB \mid BC \mid CA \mid b \mid BB \mid BCS \mid CS \mid BS \mid a \mid abC \mid ab$
$C \to SAS \mid AS \mid SA \mid SS \mid AB \mid BC \mid CA \mid b \mid BB \mid BCS \mid CS \mid BS \mid a \mid abC \mid ab$

TODOS LOS SÍMBOLOS SON ÚTILES

FORMA NORMAL DE CHOMSKY

$S \to SX \mid AS \mid SA \mid SS \mid AB \mid BC \mid CA \mid b \mid BB \mid BY \mid CS \mid BS \mid a \mid HZ \mid HI$
$A \to SX \mid AS \mid SA \mid SS \mid AB \mid BC \mid CA \mid b \mid BB \mid BY \mid CS \mid BS \mid a \mid HZ \mid HI$
$B \to SX \mid AS \mid SA \mid SS \mid AB \mid BC \mid CA \mid b \mid BB \mid BY \mid CS \mid BS \mid a \mid HZ \mid HI$
$C \to SX \mid AS \mid SA \mid SS \mid AB \mid BC \mid CA \mid b \mid BB \mid BY \mid CS \mid BS \mid a \mid HZ \mid HI$
$X \to AS \qquad Y \to CS \qquad Z \to IC \qquad H \to a \qquad I \to b$

(1.5 puntos)