Examen de Computabilidad y Complejidad

(CMC)

21 de enero de 2011

(I) CUESTIONES: (Justifique formalmente las respuestas)

1. ¿Es el lenguaje $\{x \in \{a,b,c\}^* : |x|_a \leq |x|_b \leq |x|_c\}$ incontextual?

   El lenguaje dado no es incontextual. Lo estableceremos aplicando operaciones de cierre y el lema de iteración.

   Sea $L = \{x \in \{a,b,c\}^* : |x|_a \leq |x|_b \leq |x|_c\} \cap a^*b^*c^* = \{a^i b^j c^k \mid i \leq j \leq k\}$. Puesto que la intersección con lenguajes regulares es una operación de cierre en la clase de los lenguajes incontextuales se tiene que si el lenguaje dado fuera incontextual también lo sería L.

   Seguidamente estableceremos que L no es incontextual viendo que incumple el lema de iteración. Supóngase que lo cumple y sea n la constante del lema; sea $z = a^n b^n c^n$, que pertenece a L y $|z| \geq n$, así debe poder factorizarse como z = uvwxy en las condiciones ya conocidas del lema. Veamos que esto no es posible. Cualquier posible factorización se encuentra en uno de los siguientes casos:

   I) $vwx \in a^*$, tomando una iteración $m > 1$ se obtienen palabras de la forma $t = a^{ñ} b^n c^n$ con $ñ > n$, de modo que t no pertenece a L.

   II) $vwx \in b^*$, tomando una iteración $m \neq 1$ se obtienen palabras de la forma $t = a^n b^{ñ} c^n$ con $ñ > n$ o $ñ < n$, de modo que t no pertenece a L.

   III) $vwx \in c^*$, tomando la iteración $m = 0$ se obtienen palabras de la forma $t = a^n b^n c^{ñ}$ con $ñ < n$, de modo que t no pertenece a L.

   IV) $v \in a^+b^+$ o $x \in a^+b^+$, tomando una iteración $m > 1$ se obtienen palabras t donde se mezclan las a's con las b's, de modo que t no pertenece a L.

   V) $v \in a^+$ y $x \in b^+$, tomando una iteración $m > 1$ se obtienen palabras de la forma $t = a^{ñ} b^p c^n$ con $p > n$, de modo que t no pertenece a L.

   VI) $v \in b^+c^+$ o $x \in b^+c^+$, tomando una iteración $m > 1$ se obtienen palabras t donde se mezclan las b's con las c's, de modo que t no pertenece a L.

   VII) $v \in b^+$ y $x \in c^+$, tomando la iteración $m = 0$ se obtienen palabras de la forma $t = a^n b^p c^{ñ}$ con $p < n$, de modo que t no pertenece a L.

   Puesto que no existe ninguna otra factorización, en las condiciones exigidas por el lema, se concluye que L no es incontextual y en consecuencia el lenguaje dado tampoco lo es.

   (1.5 puntos)

2. ¿Es el lenguaje $\{a^n b^m c^k : n \neq m \vee m = k\}$ incontextual?

   EL lenguaje dado puede descomponerse como $L \cup L'$, donde

   $$L = \{a^n b^m c^k / n \neq m, k \geq 0\}, \text{ y}$$
   $$L' = \{a^n b^m c^k / m = k, n \geq 0\}.$$

   Nuevamente L puede descomponerse como L_1L_2, donde

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

   También L' puede descomponerse como L_3L_4, donde

   $$L_3 = \{a^n / n \geq 0\} \text{ y } L_4 = \{b^m c^k / m = k\}.$$

   Seguidamente obtendremos para cada uno de estos lenguajes una gramática incontextual que lo genere. A partir de esto, y tomando en cuenta que los lenguajes

incontextuales son cerrados para la unión y la concatenación, tendremos establecido que el lenguaje dado es incontextual.

I) $L_1 = \{a^n b^m / n \neq m\}$. $n \neq m$ significa que $n = m + k$ o que $n + k = m$, para algún $k > 0$; en consecuencia se tiene que si $x \in L_1$, entonces

$$x = a^n b^{n+k} \quad \vee \quad x = a^{n+k} b^n, \text{ con } n \geq 0 \text{ y } k > 0.$$

En consecuencia la gramática

$$S \rightarrow aSb \mid aA \mid Bb, \quad A \rightarrow aA \mid \lambda, \quad B \rightarrow Bb \mid \lambda,$$

genera a L_1.

II) $L_2 = \{c^k / k \geq 0\}$. Este lenguaje puede, de modo inmediato, generarse con la gramática

$$S \rightarrow cS \mid \lambda.$$

III) $L_3 = \{a^n / n \geq 0\}$. Este lenguaje puede, de modo inmediato, generarse con la gramática

$$S \rightarrow aS \mid \lambda.$$

IV) $L_4 = \{b^m c^k / m = k\}$. Este lenguaje puede, de modo inmediato, generarse con la gramática

$$S \rightarrow bSc \mid \lambda.$$

(1.5 puntos)

3. Sean dos lenguajes $L_1, L_2 \subseteq \Sigma^*$, se define la operación P del modo que sigue:
$x \in P(L_1, L_2) \Leftrightarrow \exists u,v,w \in \Sigma^*: x = uvw \wedge u \in L_1 \wedge w \in L_2$.

I. ¿Es la familia de los lenguajes recursivos cerrada para la operación P?
(1.0 punto)

II. ¿ Es la familia de los lenguajes recursivamente enumerables cerrada para la operación P?
(1.0 punto)

EL lenguaje definido por $P(L_1,L_2)$ se caracteriza porque sus palabras tienen un prefijo en L_1, un sufijo en L_2, y el segmento restante en Σ^*. Así $P(L_1,L_2) = L_1\Sigma^*L_2$.

Demostraremos que las clases mencionadas en I y II son cerradas para la operación P demostrando que son cerradas para la concatenación, ya que Σ^* es un lenguaje regular y por tanto recursivo y recursivamente enumerable.

I **La clase de los lenguajes recursivos es cerrada para la concatenación**. Sean L y L', dos lenguajes recursivos; existen dos máquinas de Turing M y M' que se detienen para cada entrada con L(M) = L y L(M') = L'. Seguidamente definimos una máquina de Turing M" que se detiene para cada entrada y L(M") = LL'. M" opera como sigue.Tiene una cinta de entrada con dos sectores, en uno de ellos se escribe la palabra de entrada x y en el otro un símbolo de marcaje ✓. Este símbolo recorrerá los símbolos de xB (hasta el blanco B inmediatamente a la derecha de x). Para una posición del símbolo ✓, en el rango especificado, se factorizará x como uv, siendo u el prefijo de x que termina en el símbolo anterior al marcado y v el sufijo de x restante.
Si M" recibe como entrada x = λ, aplicará la entrada λ a M, si M rechaza M" también rechaza, si M acepta aplicará λ a M', aceptando o rechazando según ésta lo haga.
Si M" recibe como entrada x ≠ λ, pondrá ✓ sobre el primer símbolo de x y aplicará la entrada u a M, si M rechaza M" también rechaza, si M acepta aplicará v a M'; si M' acepta, M" acepta a x. Si se produce rechazo, bien por parte de M o de M', se desplaza la marca ✓ una posición a la derecha y vuelve a repetir el proceso, siempre que la marca

está dentro del rango de posiciones. Si la marca se sale del rango de posiciones M" termina rechazando a x.
Claramente M" se detiene para cualquier entrada y L(M") = LL'.

II **La clase de los lenguajes recursivamente enumerables es cerrada para la concatenación**. Sean L y L', dos lenguajes recursivamente enumerables; existen dos máquinas de Turing M y M' con L(M) = L y L(M') = L'. Seguidamente definimos una máquina de Turing M" con L(M") = LL'. M" opera como sigue. M" integra dos máquinas M y M', que reciben como entrada una palabra z y un contador j que limita el número máximo de movimientos que cada una puede realizar al procesar z simulando, respectivamente a M y a M'.
Si M" recibe como entrada x = λ, entonces en una cinta aparte inicializa el contador j a 1, y aplica λ y j a M, si ésta rechaza incrementa el contador j en uno y vuelve a repetir la operación; si acepta, aplica λ y j a M', si ésta acepta M" acepta, si no incrementa el contador j en uno y se vuelve a repetir la operación completa con el nuevo valor de j.
Si M" recibe como entrada x ≠ λ, entonces arranca un generador triangular controlado de pares (i,j), que inicialmente genera el par (1,1) y luego genera los pares sucesivos, de uno en uno, cada vez que recibe la señal de control de avance. Ante un par (i,j), si i > |x| + 1, envía al generador la señal de control, en otro caso factoriza x como uv donde u es el prefijo de x de longitud i – 1, y v es el sufijo de x restante; seguidamente aplica u y j a M, si ésta rechaza envía al generador la señal de control y vuelve a repetirse la operación con el nuevo par (i,j); si acepta, aplica v y j a M', si ésta acepta M" acepta, si no se envía al generador la señal de control y vuelve a repetirse la operación completa con el nuevo par (i,j).
Claramente L(M") = LL': $x \in LL' \Leftrightarrow (\exists u \in L)(\exists v \in L')(x = uv)$; $u \in L \Leftrightarrow u \in L(M) \Leftrightarrow (\exists n)$(M acepta a u en n pasos); $v \in L' \Leftrightarrow v \in L(M') \Leftrightarrow (\exists m)$(M' acepta a v en m pasos); sea k = máx(n,m); $x \in LL' \Leftrightarrow$ M" genera el par (|u| + 1,k); como el par (|u| + 1,k) se genera en alguna etapa se concluye que: $x \in LL' \Leftrightarrow x \in L(M")$.

(II) PROBLEMAS

4. Desarrolle un módulo *Mathematica*, adecuadamente explicado, que reciba como entrada una gramática incontextual G y devuelva el número máximo de símbolos no terminales distintos que aparecen en los consecuentes de sus producciones. Por ejemplo, para la gramática del ejercicio 6 devolvería 3.

Recorreremos secuencialmente los consecuentes de las producciones de la gramática, calcularemos el número de símbolos auxiliares distintos en cada uno de ellos y retornaremos el máximo valor de los calculados.

```
P[G_List] := Module[{i,j,producciones,consecuentes,alfa,max,
lon,aux},
  producciones = G[[3]];
  max = 0;
  For[i = 1, i ≤ Length[producciones], i++,
   consecuentes = producciones[[i,2]];
   For[j = 1, j ≤ Length[consecuentes], j++,
    alfa = consecuentes[[j]];
    aux = Intersection[alfa,G[[1]]];
    lon = Length[aux];
    If[lon > max, max = lon]
    ]
   ];
  Return[max]
  ]
```

(2.0 puntos)

5. Dada la gramática

$$G : S \rightarrow 0S1 \mid 1S \mid \lambda$$

y el homomorfismo: h(0) = 10 , h(1) = 0.
Obténgase una gramática incontextual para el lenguaje $(L(G)^R \cup h(L(G)))L(G)^*$.

$L(G)^R$: $X \rightarrow 1X0 \mid X1 \mid \lambda$
h(L(G)): $Y \rightarrow 10Y0 \mid 0Y \mid \lambda$
L(G)*: $Z \rightarrow TZ \mid \lambda$ $T \rightarrow 0T1 \mid 1T \mid \lambda$
$L(G)^R \cup h(L(G))$: $A \rightarrow X \mid Y$
$(L(G)^R \cup h(L(G)))L(G)^*$: $S \rightarrow AZ$

(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 → AS | BCD | 01 C → S0A | λ
A → ABB | 0S1A | AS | λ D → BC | D1D
B → 0B | CD

ELIMINACIÓN DE LOS SÍMBOLOS INÚTILES

Eliminación de los símbolos no generativos: B, D
S → AS | 01 C → S0A | λ A → 0S1A | AS | λ

Eliminación de los símbolos no alcanzables: C
S → AS | 01 A → 0S1A | AS | λ

ELIMINACIÓN DE LAS PRODUCCIONES-λ

Símbolos anulables: A
S → AS | 01 | S
A → 0S1A | 0S1 | AS | S

ELIMINACIÓN DE LAS PRODUCCIONES UNITARIAS

S → AS | 01
A → 0S1A | 0S1 | AS | 01

TODOS LOS SÍMBOLOS SON ÚTILES

FORMA NORMAL DE CHOMSKY

S → AS | YZ
Y → 0
Z → 1
A → YSZA | YSZ | AS | YZ

S → AS | YZ
Y → 0
Z → 1
A → YB| YD | AS | YZ
B → SC

$C \rightarrow ZA$
$D \rightarrow SZ$

(1.5 puntos)