Examen de Computabilidad y Complejidad

(CMC)

11 de septiembre de 2001

(I) Cuestiones (justifique formalmente las respuestas)

1. Sea el alfabeto Σ y la colección de lenguajes definidos sobre el mismo $\{L_i\}_{i=1}^n$ tales que $(\forall i \neq j)\ L_i \cap L_j = \emptyset$ y $\bigcup_{i=1}^n L_i = \Sigma^*$. Si todos los lenguajes L_i son recursivamente enumerables ¿ son todos los lenguajes L_i recursivos ?

   (1.5 ptos)

   Solución

   Tomemos el valor n constante. Es fácil comprobar que para cualquier lenguaje L_i su lenguaje complementario toma la siguiente forma $\overline{L_i} = \bigcup_{k=1}^{i-1} L_k \cup \bigcup_{k=i+1}^{n} L_k$. Por otra parte, la unión finita de lenguajes recursivamente enumerables es, a su vez, un lenguaje recursivamente enumerable. Así, tanto L_i como $\overline{L_i}$ son lenguajes recursivamente enumerables y, a partir de un resultado visto en clase, son recursivos.

2. Sea el lenguaje $L = \{w\#x : x \in \{a,b\}^* \text{ y } w \text{ es un segmento de } x\}$. ¿ Es el lenguaje L incontextual ?

   (1.5 ptos)

   Solución

   El lenguaje L no es incontextual. Resolveremos esta cuestión mediante el lema de bombeo. Sea n la constante del lema. Supongamos que la cadena $z = a^n b^n \# a^n b^n = uvwxy$ cumple las dos condiciones iniciales del lema y veamos que la tercera condición no se cumple. Para ello comprobaremos que no $\forall i \geq 0\ uv^iwx^iy \in L$. Haremos un estudio por casos

   (a) vx formado por símbolos a del primer bloque y de forma que $|vx| = k \geq 1$
   En este caso, podemos tomar un valor $i = 2$ y formamos la cadena $a^{n+k}b^n\#a^nb^n$ que no pertenece al lenguaje ya que la primera subcadena no es segmento de la segunda.

   (b) vx formado por símbolos b del primer bloque y de forma que $|vx| = k \geq 1$
   Tomamos de nuevo el valor $i = 2$ y formamos la cadena $a^nb^{n+k}\#a^nb^n$ que, al igual que en el caso anterior, no pertenece al lenguaje.

   (c) vx formado por símbolos a del segundo bloque y de forma que $|vx| = k \geq 1$
   Aquí tomaremos el valor $i = 0$ y formamos la cadena $a^nb^n\#a^{n-k}b^n$ que tampoco pertenece al lenguaje.

   (d) vx formado por símbolos b del segundo bloque y de forma que $|vx| = k \geq 1$
   Como en el caso anterior, si tomamos $i = 0$ entonces la nueva cadena $a^nb^n\#a^nb^{n-k}$ no pertenece al lenguaje.

(e) vx contiene al símbolo #

Tomando $i = 2$ entonces la cadena $uvvwxxy$ contiene dos símbolos # y no pertenece al lenguaje.

(f) vx formado por símbolos a y b del primer bloque de forma que $|vx|_a = j$ y $|vx|_b = k$ con $j, k \geq 1$

Tomando el valor $i = 2$ entonces la nueva cadena toma la forma $a^{n+j}b^{n+k}\#a^nb^n$ que no pertenece al lenguaje.

(g) vx formado por símbolos a y b del segundo bloque de forma que $|vx|_a = j$ y $|vx|_b = k$ con $j, k \geq 1$

Tomando el valor $i = 0$ entonces la nueva cadena toma la forma $a^nb^n\#a^{n-j}b^{n-k}$ que no pertenece al lenguaje.

(h) vx formado por símbolos b del primer bloque y a del segundo bloque de forma que $|vx|_b = j$ y $|vx|_a = k$ con $j, k \geq 1$

Tomemos el valor de $i = 2$ y formamos la cadena $a^nb^{n+j}\#a^{n+k}b^n$ que, como en los casos anteriores, no pertenece al lenguaje.

Debido a la condición del lema ya no se pueden plantear más casos y, al haber demostrado en todos los casos posibles, que el lema no se cumple en su tercera condición, entonces podemos concluir que el lenguaje no es incontextual.

3. Sea $\sigma : \Sigma \to \mathcal{P}(\Sigma^*)$ una sustitución de forma que para todo símbolo a, $\sigma(a)$ es un lenguaje recursivamente enumerable. Pronúnciese acerca de la veracidad o falsedad de las siguientes afirmaciones

(a) Si L es recursivo entonces $\sigma(L)$ también lo es

(b) Si L es recursivamente enumerable pero no recursivo entonces $\sigma(L)$ también es recursivamente enumerable pero no recursivo

(2 ptos)

Solución

(a) El enunciado es falso. Tomemos la sustitució σ de forma que $\forall a \in \Sigma \ \sigma(a) = L_U$ siendo L_U el lenguaje universal visto en clase que es recursivamente enumerable pero no recursivo. Si ahora tomamos el lenguaje $L = \{a\}$ es fácil comprobar que L es recursivo ya que está formado por una única cadena y, por lo tanto, al ser finito se puede construir trivialmente una máquina de Turing que lo acepte y siempre pare. La sustitución sobre L nos da el resultado $\sigma(L) = L_U$. Por lo tanto, L es recursivo y $\sigma(L)$ no lo es.

(b) El enunciado es falso. Tomemos en este caso la sustitución σ de forma que $\forall a \in \Sigma \ \sigma(a) = \emptyset$ (el lenguaje vacío). El lenguaje vacío es recursivo (por lo tanto es recursivamente enumerable) ya que es aceptado por una máquina de Turing que para siempre rechazando su entrada. Basta con no definir ningún movimiento y hacer que el estado inicial no sea final. Tomamos ahora el lenguaje recursivamente enumerable no recursivo L_U del apartado anterior. Recordemos que L_U es recursivamente enumerable pero no recursivo. Si aplicamos la sustitución definida entonces tenemos que $\sigma(L_U) = \emptyset$. Por lo tanto, L_U es recursivamente enumerable pero no recursivo y $\sigma(L_U)$ es recursivo.

(II) PROBLEMAS:

4. Dada una gramática incontextual, diremos que una producción es *recursiva* si el símbolo auxiliar de la parte izquierda de la producción también aparece en su parte derecha. Se pide escribir un módulo *Mathematica* que, dada una gramática incontextual como parámetro de entrada, devuelva la misma gramática pero sin reglas recursivas.

(2 ptos)

Solución

```
problema4[G_List]:= Module[ { Aux,Ter,P,Axioma,NuevaP,NAP,i,j,dcha},
        Aux = G[[1]];
        Ter = G[[2]];
        P = G[[3]];
        Axioma = G[[4]];
        NuevaP = {};
        For[ i=1, i≤ Length[P],i++,
                NAP = {};
                AppendTo[NAP,P[[i,1]]];
                dcha = {};
                For[ j=1, j≤ Length[P[[i,2]]], j++,
                        If[!MemberQ[P[[i,2,j]],P[[i,1]]], AppendTo[dcha,P[[i,2,j]]]]
                ];
                If[Length[dcha] > 0,
                        AppendTo[NAP,dcha];
                        AppendTo[NuevaP,NAP]
                ]
        ];
        Return[{ Aux, Ter, NuevaP, Axioma }]
]
```

5. Sea la gramática G definida por las producciones $S \to AS \mid 0$; $A \to BB \mid 1$; $B \to 0S1 \mid \lambda$. Sea el homomorfismo h tal que $h(0) = a$ y $h(1) = bb$. Se pide construir una gramática incontextual que genere el lenguaje $h(L(G)^* \cup (L(G))^r)$.

(1 pto)

Solución

Actuaremos aplicando cada una de las operaciones por su orden de precedencia. En primer lugar, definimos una gramática para $L(G)^*$ mediante las siguientes reglas donde ya se han renombrado los símbolos auxiliares de G

$S_0 \to S'S_0 \mid \lambda$

$S' \to A'S' \mid 0$

$A' \to B'B' \mid 1$

$B' \to 0S'1 \mid \lambda$

A continuación obtenemos una gramática para $(L(G))^r$ renombrando de nuevo los símbolos auxiliares de G

$S'' \to S''A'' \mid 0$

$A'' \to B''B'' \mid 1$

$B'' \to 1S''0 \mid \lambda$

Obtenemos, a partir de las dos anteriores gramáticas, una nueva gramática para el lenguaje $L(G)^* \cup (L(G))^r$

$S_U \to S_0 \mid S''$
$S_0 \to S'S_0 \mid \lambda \quad S'' \to S''A'' \mid 0$
$S' \to A'S' \mid 0 \quad A'' \to B''B'' \mid 1$
$A' \to B'B' \mid 1 \quad B'' \to 1S''0 \mid \lambda$
$B' \to 0S'1 \mid \lambda.$

Por último, aplicando el homomorfismo definido en el problema sobre la anterior gramática, obtenemos un gramática que genera el lenguaje $h(L(G)^* \cup (L(G))^r)$. La gramática que se nos pide queda definida por las siguientes reglas

$S_U \to S_0 \mid S''$
$S_0 \to S'S_0 \mid \lambda \quad S'' \to S''A'' \mid a$
$S' \to A'S' \mid a \quad A'' \to B''B'' \mid bb$
$A' \to B'B' \mid bb \quad B'' \to bbS''a \mid \lambda$
$B' \to aS'bb \mid \lambda.$

6. Dada la gramática G definida por las siguientes producciones se pide obtener una gramática simplificada y en Forma Normal de Chomsky que genere $L(G) - \{\lambda\}$

$S \to CA \mid AEF \mid 0F \quad D \to SS \mid \lambda$
$A \to BB \mid \lambda \quad E \to CD \mid SFA \mid 1$
$B \to BD \mid 0C1 \quad F \to BE \mid EEE \mid AS \mid A$
$C \to DC \mid 0B$

(2 ptos)

Solución

En primer lugar procederemos a simplificar la gramática G.

Eliminación de símbolos no generativos

Símbolos no generativos: $\{B, C\}$

Gramática sin símbolos no generativos:

$S \to AEF \mid 0F \quad D \to SS \mid \lambda$
$A \to \lambda \quad E \to SFA \mid 1$
$F \to EEE \mid AS \mid A$

Eliminación de símbolos no alcanzables

Símbolos no alcanzables: $\{D\}$

Gramática sin símbolos no alcanzables:

$S \to AEF \mid 0F \quad A \to \lambda$
$E \to SFA \mid 1 \quad F \to EEE \mid AS \mid A$

Eliminación de producciones vacías

Símbolos anulables: $\{A, F\}$

Gramática sin producciones vacías:

$S \to AEF \mid EF \mid AE \mid E \mid 0F \mid 0$
$E \to SFA \mid SA \mid SF \mid S \mid 1 \quad F \to EEE \mid AS \mid S \mid A$

Eliminación de producciones unitarias

$C(S) = \{S, E\}$, $C(A) = \{A\}$, $C(E) = \{E, S\}$ y $C(F) = \{F, S, E, A\}$

Gramática sin producciones unitarias:

$S \to AEF \mid EF \mid AE \mid SFA \mid SA \mid SF \mid 1 \mid 0F \mid 0$

$E \to SFA \mid SA \mid SF \mid AEF \mid EF \mid AE \mid 0F \mid 0 \mid 1$

$F \to EEE \mid AS \mid AEF \mid EF \mid AE \mid SFA \mid SA \mid SF \mid 1 \mid 0F \mid 0$

Volvemos a calcular de nuevo los símbolos inútiles.

<u>Eliminación de símbolos no generativos</u>

Símbolos no generativos: $\{A\}$

Gramática sin símbolos no generativos:

$S \to EF \mid SF \mid 1 \mid 0F \mid 0$

$E \to SF \mid EF \mid 0F \mid 0 \mid 1$

$F \to EEE \mid EF \mid SF \mid 1 \mid 0F \mid 0$

<u>Eliminación de símbolos no alcanzables</u>

Símbolos no alcanzables: $\emptyset$

Por lo tanto, la gramática simplificada es la definida por las siguientes reglas

$S \to EF \mid SF \mid 1 \mid 0F \mid 0$

$E \to SF \mid EF \mid 0F \mid 0 \mid 1$

$F \to EEE \mid EF \mid SF \mid 1 \mid 0F \mid 0$

Procedemos a continuación a normalizar la anterior gramática.

<u>Obtención de una gramática en Forma Normal de Chomsky</u>

$S \to EF \mid SF \mid 1 \mid C_0F \mid 0$

$E \to SF \mid EF \mid C_0F \mid 0 \mid 1$

$F \to EEE \mid EF \mid SF \mid 1 \mid C_0F \mid 0$

$C_0 \to 0$

Factorización de las producciones y gramática final en FNC

$S \to EF \mid SF \mid 1 \mid C_0F \mid 0$

$E \to SF \mid EF \mid C_0F \mid 0 \mid 1$

$F \to ED_1 \mid EF \mid SF \mid 1 \mid C_0F \mid 0$

$D_1 \to EE$

$C_0 \to 0$