Examen de Computabilidad y Complejidad

(CMC)

25 de junio de 2001

(I) Cuestiones (justifique formalmente las respuestas)

1. ¿ Son incontextuales los siguientes lenguajes ?

   (a) $L_1 = \{a^n b^n c^i | n \le i \le 2n\}$

   (b) $L_2 = \{a^n b^m | m, n \ge 0, (m = n) \vee (m = 2n)\}$

(2 ptos)

Solución

(a) El lenguaje $L_1 = \{a^n b^n c^i | n \le i \le 2n\}$ no es incontextual. La demostración la haremos mediante el lema de bombeo. Tomemos k como la constante del lema y $z = a^k b^k c^k \in L$. Supongamos que la cadena z cumple las dos condiciones iniciales del lema y veremos que la tercera condición no se cumple. Para ello, tomemos $z = uvwxy = a^k b^k c^k$ y veamos que no $\forall i \ge 0\ uv^i wx^i y \in L$. Haremos un estudio por casos

   i. vx formado sólo por símbolos a y $|vx| = j \ge 1$
   Tomando $i = 0$ se forma la cadena $a^{k-j} b^k c^k$ que no pertenece a L ya que el número de as es distinto del número de bs.

   ii. vx formado sólo por símbolos b y $|vx| = j \ge 1$
   Tomando $i = 0$ se forma la cadena $a^k b^{k-j} c^k$ que no pertenece a L ya que el número de as es distinto del número de bs.

   iii. vx formado sólo por símbolos c y $|vx| = j \ge 1$
   Tomando $i = 0$ se forma la cadena $a^k b^k c^{k-j}$ que no pertenece a L ya que el número de cs es menor que el de as y que el de bs.

   iv. vx formado por los símbolos a y b. Supongamos $|vx|_a = j \ge 1$ y $|vx|_b = p \ge 1$
   Tomando $i = 2$ se forma la cadena $uvvwxxy$ que no pertenece a L ya que el número de cs es menor que el de as (que es $k + j$) y que el de bs (que es $k + p$).

   v. vx formado por los símbolos b y c. Supongamos $|vx|_b = j \ge 1$ y $|vx|_c = p \ge 1$
   Tomando $i = 0$ se forma la cadena $a^k b^{k-j} c^{k-p}$ que no pertenece a L ya que el número de cs es menor que el de as y el número de bs es distinto del número de as.

Puesto que en todos los casos que se han podido plantear sobre la cadena z se ha verificado que la condición (3) del lema de bombeo no se cumple, entonces L no es incontextual.

(b) El lenguaje $L_2 = \{a^n b^m | m, n \geq 0, (m = n) \vee (m = 2n)\}$ se puede generar con la siguiente gramática incontextual

$S \rightarrow A \mid B \mid \lambda$

$A \rightarrow aAb \mid ab$

$B \rightarrow aBbb \mid abb$

Por lo tanto L_2 es incontextual.

2. Sea el alfabeto $\Sigma = \{a, b\}$ y la operación P definida sobre cadenas de Σ de la forma $\mathrm{P}(x) = c^{2|x|}x \ \forall x \in \Sigma^*$. Se extiende la operación a lenguajes de la forma habitual.

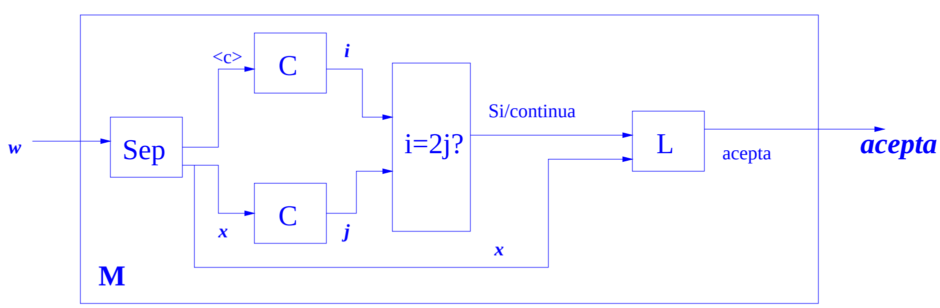
   (a) ¿ Es la clase de los lenguajes recursivamente enumerables cerrada respecto de P ?

   (b) ¿ Y la clase de los lenguajes recursivos ? (2 ptos)

Solución

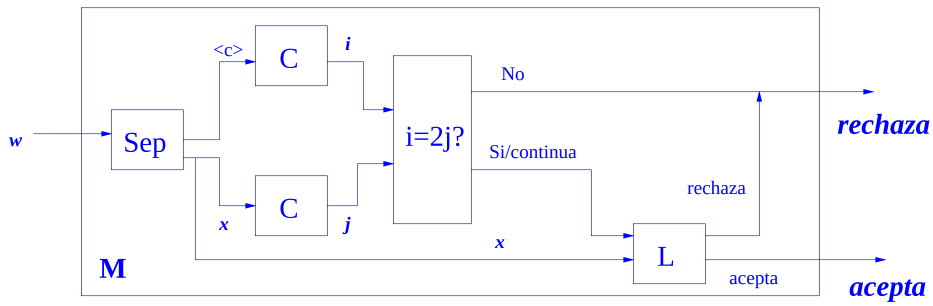
(a) La clase $\mathcal{L}_{re}$ es cerrada bajo P. Para comprobarlo, construiremos una MT que acepte $\mathrm{P}(L)$, siendo L un lenguaje recursivamente enumerable.

Contamos con una máquina L que acepta L. Contamos también con un módulo Sep que separa la cadena de entrada en dos cadenas: la cadena $< c >$ que está formada por el prefijo formado por símbolos c y x que es el sufijo de la entrada. El módulo C devuelve la longitud de una cadena de entrada. Por último, el módulo $i = 2j?$ es un comparador de números enteros. Para cada uno de estos módulos se pueden construir máquinas que responden a algoritmos con salida siempre definida. A partir de L, Sep, C y $i = 2j?$ podemos construir el siguiente esquema de una máquina de Turing que acepta $\mathrm{P}(L)$



La máquina de la figura anterior funciona de la siguiente forma: Inicialmente el módulo Sep separa la cadena de entrada en las cadenas $< c >$ y x de acuerdo con el criterio anterior. El módulo C cuenta las longitudes de las dos cadenas dando como salida los valor i y j respectivamente. El módulo $i = 2j?$ establece si el valor de i es igual al doble de j. Si este módulo contesta afirmativamente, entonces la cadena x se le pasa al módulo L para comprobar si pertenece o no al lenguaje L. Si la máquina L acepta la cadena x, entonces la cadena de entrada tiene la forma $c^{2|x|}x$ donde x pertenece a L y por lo tanto se acepta ya que pertenece a $\mathrm{P}(L)$.

(b) La clase $\mathcal{L}_{rec}$ es cerrada bajo P. Para comprobarlo, construiremos una MT que acepte P(L), siendo L un lenguaje recursivo, y que pare ante cualquier entrada. Contamos con los mismos módulos que en el caso anterior: Sep, C, $i = 2j$? y L. Para cada uno de estos módulos se pueden construir máquinas que responden a algoritmos con salida siempre definida. A partir de ellos podemos construir el siguiente esquema de una máquina de Turing que acepta P(L)



La máquina de la figura anterior funciona de la siguiente forma: Inicialmente el módulo Sep separa la cadena de entrada en las cadenas $< c >$ y x de acuerdo con el criterio anterior. El módulo C cuenta las longitudes de las dos cadenas dando como salida los valor i y j respectivamente. El módulo $i = 2j$? establece si el valor de i es igual al doble de j. Si este módulo contesta afirmativamente, entonces la cadena x se le pasa al módulo L. Si el módulo $i = 2j$? contesta negativamente entonces la cadena de entrada se rechaza, ya que no toma la forma de la operación P. Para comprobar si la cadena x pertenece o no al lenguaje L contamos con la máquina L que, en tiempo finito, establece la condición de pertenencia. Si la máquina L acepta la cadena x, entonces la cadena de entrada tiene la forma $c^{2|x|}x$ donde x pertenece a L y por lo tanto se acepta ya que pertenece a P(L). En caso contrario se rechaza la cadena de entrada.

3. Pronúnciese acerca de la veracidad o falsedad de la siguiente afirmación : *"Si M es una máquina de Turing que acepta un lenguaje incontextual entonces M para ante cualquier cualquier cadena que se le proporcione como entrada"*. (1 pto)

   Solución

   La afirmación es falsa. Daremos el siguiente contraejemplo definiendo la máquina de Turing M con $\Sigma = \{a, b\}$, $\Gamma = \{a, b, B\}$, $Q = \{q_0, q_1, q_2\}$, $F = \{q_1\}$ y los siguientes movimientos en la función δ

   $\delta(q_0, a) = (q_1, a, R)$

   $\delta(q_0, b) = (q_2, b, R)$

   $\delta(q_2, a) = \delta(q_2, b) = \delta(q_2, B) = (q_2, B, R)$

   Es fácil comprobar que el lenguaje que acepta esta máquina es el formado por las cadenas que comienzan por a. Es decir, $L(M) = \{aw | w \in \{a, b\}^*\}$. De igual forma, es fácil comprobar que M no para al procesar las cadenas que comienzan por b. Por último, $L(M)$ es incontextual ya que se puede generar con la gramática definida por las producciones $S \to aA$; $A \to aA \mid bA \mid \lambda$.

(II) PROBLEMAS:

4. Dada una gramática incontextual, diremos que un símbolo auxiliar A es *recursivo por la izquierda* si la gramática contiene alguna producción con la forma $A \to A\alpha$.

Se pide escribir un módulo *Mathematica* que, dada una gramática incontextual como parámetro de entrada, devuelva un conjunto (en formato de lista) que contenga los símbolos no terminales recursivos por la izquierda de la gramática. (2 ptos)

Solución

```
Recursivos[G_List]:=Module[{ P, Solucion, k, izda, dchas, j },
    P=G[[3]];
    Solucion={};
    For[k=1, k ≤ Length[P], k++,
        izda = P[[k,1]];
        dchas = P[[k,2]];
        For[j=1, j ≤ Length[dchas], j++,
            If[Length[dchas[[j]]] > 0,
                If[First[dchas[[j]]] == izda,
                    Solucion = Union[Solucion, { First[dchas[[j]]] } ]
                ]
            ]
        ]
    ];
    Return[Solucion]
]
```

5. Sea la gramática G definida por las producciones $S \to 0A \mid 1S1$; $A \to S00 \mid 1$. Sea el homomorfismo h tal que $h(0) = a$ y $h(1) = ba$. Se pide construir una gramática incontextual que genere el lenguaje $h(L(G))((L(G))^r \cup L(G))$.

(1 pto)

Solución

En primer lugar construimos una gramática para $h(L(G))$

$S' \to aA' \mid baS'ba$

$A' \to S'aa \mid ba$

A continuación una gramática para $(L(G))^r$

$S'' \to A''0 \mid 1S''1$

$A'' \to 00S'' \mid 1$

Una gramática para $((L(G))^r \cup L(G))$

$S_0 \to S'' \mid S$

$S'' \to A''0 \mid 1S''1$

$A'' \to 00S'' \mid 1$

$S \to 0A \mid 1S1$

$A \to S00 \mid 1$

Y por último una gramática para $h(L(G))((L(G))^r \cup L(G))$

$S_c \to S'S_0$

$S' \to aA' \mid baS'ba$

$A' \to S'aa \mid ba$

$S_0 \to S'' \mid S$

$S'' \to A''0 \mid 1S''1$

$A'' \to 00S'' \mid 1$

$S \to 0A \mid 1S1$

$A \to S00 \mid 1$

El axioma de esta gramática se corresponde con S_c.

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

   $S \to BD \mid 0A1A$

   $A \to 0 \mid CS \mid SS \mid \lambda$

   $B \to 00B \mid BS \mid ABD$

   $C \to AA \mid CA \mid 0B$

   $D \to 0S \mid A1C$ (2 ptos)

   Solución

   En primer lugar procederemos a simplificar la gramática.

   Eliminación de símbolos no generativos

   Símbolos no generativos: $\{B\}$

   Gramática sin símbolos no generativos

   $S \to 0A1A$

   $A \to 0 \mid CS \mid SS \mid \lambda$

   $C \to AA \mid CA$

   $D \to 0S \mid A1C$

   Eliminación de símbolos no alcanzables

   Símbolos no alcanzables: $\{D\}$

   Gramática sin símbolos no alcanzables

   $S \to 0A1A$

   $A \to 0 \mid CS \mid SS \mid \lambda$

   $C \to AA \mid CA$

   Eliminación de producciones vacías

   Símbolos anulables: $\{A, C\}$

   Gramática sin producciones vacías

   $S \to 0A1A \mid 01A \mid 0A1 \mid 01$

   $A \to 0 \mid CS \mid S \mid SS$

   $C \to AA \mid A \mid CA \mid C$

   Eliminación de producciones unitarias

   $C(S) = \{S\}$, $C(A) = \{A, S\}$, $C(C) = \{C, A, S\}$

   Gramática sin producciones unitarias

   $S \to 0A1A \mid 01A \mid 0A1 \mid 01$

   $A \to 0 \mid CS \mid 0A1A \mid 01A \mid 0A1 \mid 01 \mid SS$

   $C \to AA \mid 0 \mid CS \mid 0A1A \mid 01A \mid 0A1 \mid 01 \mid SS \mid CA$

   Volvemos a comprobar los símbolos inútiles (no generativos y no alcanzables) y la gramática anterior está totalmente simplificada.

   Obtención de una gramática en forma normal de Chomsky

$S \rightarrow C_0AC_1A \mid C_0C_1A \mid C_0AC_1 \mid C_0C_1$

$A \rightarrow 0 \mid CS \mid C_0AC_1A \mid C_0C_1A \mid C_0AC_1 \mid C_0C_1 \mid SS$

$C \rightarrow AA \mid 0 \mid CS \mid C_0AC_1A \mid C_0C_1A \mid C_0AC_1 \mid C_0C_1 \mid SS \mid CA$

$C_0 \rightarrow 0$

$C_1 \rightarrow 1$

Factorización de las producciones y gramática final en FNC

$S \rightarrow C_0D_1 \mid C_0D_2 \mid C_0D_3 \mid C_0C_1$

$D_1 \rightarrow AD_2$

$D_2 \rightarrow C_1A$

$D_3 \rightarrow AC_1$

$A \rightarrow 0 \mid CS \mid C_0D_1 \mid C_0D_2 \mid C_0D_3 \mid C_0C_1 \mid SS$

$C \rightarrow AA \mid 0 \mid CS \mid C_0D_1 \mid C_0D_2 \mid C_0D_3 \mid C_0C_1 \mid SS \mid CA$

$C_0 \rightarrow 0$

$C_1 \rightarrow 1$