Examen de Computabilidad y Complejidad

(CMC)

17 de junio de 1999

(I) Cuestiones (justifique formalmente las respuestas)

1. Sea $L = \{xx^r x : x \in (a+b)^*\}$. ¿ Es L incontextual ?

   (1.5 ptos)

   Solución

   El lenguaje L no es incontextual. Para demostrarlo, partiremos de L y, mediante aplicación de propiedades de cierre, llegaremos a un lenguaje no incontextual.

   En primer lugar tomemos $L_1 = L \cap ba^*bba^*bba^*b = \{ba^nbba^nbba^nb : n \geq 0\}$. Tomemos ahora el homomorfismo g definido como $g(a) = a$, $g(b) = a$, $g(c) = a$ y $g(d) = b$. Se cumple que $g^{-1}(L_1) = L_2 = \{d\{a,b,c\}^n dd\{a,b,c\}^n dd\{a,b,c\}^n d : n \geq 0\}$. A continuación, tomamos $L_3 = L_2 \cap da^*ddb^*ddc^*d = \{da^nddb^nddc^nd : n \geq 0\}$. Por último, definimos el homomorfismo h de tal que $h(a) = a$, $h(b) = b$, $h(c) = c$ y $h(d) = \lambda$ y obtenemos $h(L_3) = \{a^nb^nc^n : n \geq 0\}$ que, como es bien sabido, no es incontextual. Como conclusión, podemos afirmar que L no es incontextual ya que, si lo fuera, el resultado que hemos obtenido, tras aplicar operaciones de cierre, también lo sería.

2. Una gramática lineal es una gramática incontextual tal que el número de no terminales en la parte derecha de sus reglas es menor o igual que uno. Un lenguaje es lineal si es generado por una gramática lineal. Sea $\mathcal{L}$ la clase de los lenguajes lineales.

   (a) Demostrar que $\mathcal{L}$ es cerrada bajo unión de lenguajes.

   (b) ¿ Es $\mathcal{L}$ cerrada bajo intersección ?

   (c) ¿ Es $\mathcal{L}$ cerrada bajo complementación ?

   (1.5 ptos)

   Solución

   Contestaremos a cada una de las cuestiones por separado.

   (a) Para demostrar que $\mathcal{L}$ es cerrada bajo unión propondremos un algoritmo constructivo que, dadas dos gramáticas lineales $G_1 = (N_1, \Sigma_1, P_1, S_1)$ y $G_2 = (N_2, \Sigma_2, P_2, S_2)$, con $N_1 \cap N_2 = \emptyset$, construya una gramática lineal G de forma que $L(G) = L(G_1) \cup L(G_2)$. La definición de G es como sigue $G = (N_1 \cup N_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S \to S_1 | S_2\}, S)$, donde $S \notin N_1 \cup N_2$. Obsérvese que G es lineal, ya que todas sus producciones tienen, a la parte derecha, como máximo un sólo símbolo auxiliar. Por otra parte, $L(G) = L(G_1) \cup L(G_2)$ ya que las derivaciones en G se forman a partir de las de G_1 o las de G_2. La construcción que hemos proporcionado coincide con la que en su momento se propuso para la unión de gramáticas incontextuales.

Puesto que, dados dos lenguajes lineales L_1 y L_2 descritos a partir de sus gramáticas, siempre podemos construir una gramática lineal que genera su unión, la conclusión a la que llegamos es que la unión es una propiedad de cierre para la clase $\mathcal{L}$.

(b) La clase $\mathcal{L}$ no es cerrada bajo intersección. A modo de contraejemplo, tomemos el lenguaje $L_1 = \{a^i b^i c^j : i, j \geq 0\}$ que es lineal al ser generado por la gramática lineal definida por las producciones $S_1 \rightarrow S_1 c | A_1 | \lambda$; $A_1 \rightarrow a A_1 b | \lambda$. Tomemos también el lenguaje $L_2 = \{a^i b^j c^j : i, j \geq 0\}$ que es lineal al ser generado por la gramática lineal definida por las producciones $S_2 \rightarrow a S_2 | A_2 | \lambda$; $A_2 \rightarrow b A_2 c | \lambda$. Al efectuar la intersección $L_1 \cap L_2$ obtenemos el lenguaje $\{a^i b^i c^i : i \geq 0\}$ que no es lineal ya que, tal y como se ha visto en repetidas ocasiones, no es incontextual.

(c) La clase $\mathcal{L}$ no es cerrada bajo complementación. Tal y como se ha visto en el anterior punto, la clase $\mathcal{L}$ no es cerrada bajo intersección. Demostraremos que si $\mathcal{L}$ fuera cerrada bajo complementación entonces también lo sería bajo intersección. Tomemos dos lenguajes lineales arbitrarios L_1 y L_2 y supongamos que la clase $\mathcal{L}$ es cerrada bajo complementario. Entonces $\overline{L_1}$ y $\overline{L_2}$ serían lineales. De igual forma $\overline{L_1} \cup \overline{L_2}$ sería lineal ya que $\mathcal{L}$ es cerrada bajo unión. Suponiendo que la complementación es de cierre, entonces $\overline{\overline{L_1} \cup \overline{L_2}} = \overline{\overline{L_1}} \cap \overline{\overline{L_2}} = L_1 \cap L_2$ también sería lineal, sean cuales fueren L_1 y L_2, lo cual hemos demostrado en el punto anterior que es falso. Por lo tanto, la complementación no puede ser una operación de cierre para la clase de los lenguajes lineales.

3. Sea L un lenguaje recursivo. Sea $L' = \{a^{|x|} b^{|y|} : x \in L \wedge y \notin L\}$. ¿ Es L' recursivo ?

(1.5 ptos)

Solución

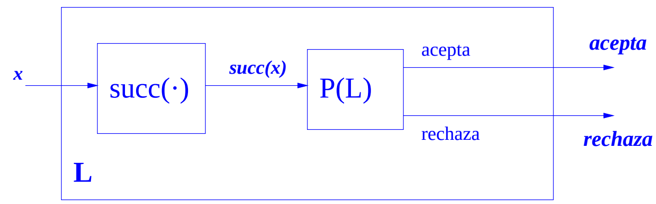
Supongamos que $L \subseteq \Sigma^*$ y asumimos que la clase de los lenguajes recursivos es cerrada bajo complementario, concatenación y homomorfismos alfabéticos (ésto es homomorfismos que sustituyen cada símbolo del alfabeto por un sólo símbolo). Definamos a continuación el homomorfismo h de forma que para todo símbolo $c \in \Sigma$ $h(c) = a$ y el homomorfismo g de forma que para todo símbolo $c \in \Sigma$ $g(c) = b$. Es fácil comprobar que $L' = h(L)g(\overline{L})$. Al poder expresar L' como el resultado de aplicar, sobre un lenguaje recursivo, operaciones de cierre para la clase de los lenguajes recursivos, podemos afirmar que L' es recursivo.

4. Sea L un lenguaje recursivamente enumerable no recursivo. Se define $P(L) = \{x : pred(x) \in L\}$, donde $pred(x)$ es predecesor de x en orden lexicográfico ($pred(\lambda) = \lambda$). ¿ Es $P(L)$ recursivo ?

(1.5 ptos)

Solución

$P(L)$ no puede ser recursivo. Suponiendo que $P(L)$ fuera recursivo, entonces L también lo sería. Actuemos por el método de contradicción y supongamos que $P(L)$ es recursivo. Podemos utilizar la función $succ(x)$ que, dada la cadena x calcula su sucesor en orden lexicográfico. A partir del módulo $succ(\cdot)$ y del esquema recursivo para $P(L)$ podemos construir un esquema recursivo para L que se muestra a continuación



El funcionamiento del anterior esquema es sencillo: Dada una cadena de entrada x se calcula su sucesor en orden lexicográfico mediante el módulo $succ(\cdot)$ y se pasa como entrada al módulo $P(L)$. Si $P(L)$ acepta $succ(x)$ entonces la máquina acepta x ya que $pred(succ(x)) = x \in L$. Por otra parte, si $P(L)$ rechaza $succ(x)$ entonces la máquina rechaza x ya que $pred(succ(x)) = x \notin L$.

Como se puede comprobar, el anterior esquema que hemos propuesto hace que L sea un lenguaje recursivo. Dado que el funcionamiento recursivo del anterior esquema sólo se puede justificar a partir de que $P(L)$ sea recursivo, la conclusión es que $P(L)$ no es recursivo ya que L tampoco lo es.

(II) PROBLEMAS:

5. Sea G la gramática:

   $S \to aA \mid bS$

   $A \to BA \mid aA \mid b$

   $B \to Bb \mid b$

   $C \to AB \mid BA \mid a$

   Obtener una gramática en forma normal de Chomsky para $(L(G))^* - \{\lambda\}$.

   (2 ptos)

   Solución

   En primer lugar, simplificaremos G para facilitar la obtención de la gramática en Forma Normal de Chomsky.

   Eliminación de símbolos no generativos

   Símbolos no generativos: $\{\}$

   Gramática sin símbolos no generativos

   $S \to aA \mid bS$

   $A \to BA \mid aA \mid b$

   $B \to Bb \mid b$

   $C \to AB \mid BA \mid a$

   Eliminación de símbolos no alcanzables

   Símbolos no alcanzables: $\{C\}$

   Gramática sin símbolos no alcanzables

   $S \to aA \mid bS$

   $A \to BA \mid aA \mid b$

   $B \to Bb \mid b$

   Eliminación de producciones vacías

   La anterior gramática no tiene producciones vacías.

   Eliminación de producciones unitarias

   La anterior gramática no tiene producciones unitarias y, además, ya está totalmente simplificada puesto que todos sus símbolos son útiles.

Pasamos, a continuación, a obtener una gramática equivalente a la gramática simplificada que esté en Forma Normal de Chomsky

Paso a Forma Normal de Chomsky

Sustitución de símbolos terminales

$S \to C_a A \mid C_b S$

$A \to BA \mid C_a A \mid b$

$B \to BC_b \mid b$

$C_a \to a$

$C_b \to b$

Es fácil comprobar que la anterior gramática ya está en Forma Normal de Chomsky y es la gramática que se solicitaba en el enunciado del problema.

6. Escríbase un módulo *Mathematica* que, dada una gramática, devuelva cierto o falso según la gramática sea lineal o no (vease la definición de gramática lineal en la cuestión segunda). Se asume que la gramática se expresa de la manera explicada en clase de práticas.

(2 ptos)

Solución

```
Solucion[G_List]:=Module[{ P, test, k, j, auxi, numauxi, l },
    P=G[[3]];
    auxi=G[[1]];
    test=True;
    k=1;
    While[test && k≤Length[P],
        j=1;
        While[test && j≤Length[P[[k,2]]],
            numauxi = 0;
            For[l=1, l ≤ Length[P[[k,2,j]]], l++,
                If[MemberQ[auxi,P[[k,2,j,l]]], numauxi++]
            ];
            If[numauxi ≥ 2, test=False];
            j++
        ];
        k++
    ];
    Return[test]
]
```