Examen de Computabilidad y Complejidad

(CMC)

16 de junio de 1998

(I) Cuestiones (justifique formalmente las respuestas)

1. Sea $L = \{0^k 1 0^{2k} 1 0^{3k} \mid k \geq 0\}$. ¿ Es L incontextual ?

(1 pto)

Solución

El lenguaje L no es incontextual. Partiremos del lenguaje L y, mediante operaciones de cierre para la clase de los lenguajes incontextuales, llegaremos a un lenguaje que no sea incontextual.

En primer lugar, definimos un homomorfismo h de forma que $h(a) = 0$, $h(b) = 0$, $h(c) = 0$ y $h(d) = 1$ y $L_1 = h^{-1}(L) = \{\{a,b,c\}^k d \{a,b,c\}^{2k} d \{a,b,c\}^{3k} \mid k \geq 0\}$. A continuación, definimos $L_2 = L_1 \cap a^* d b^* d c^* = \{a^k d b^{2k} d c^{3k} \mid k \geq 0\}$. Definimos el homomorfismo g de forma que $g(a) = a$, $g(b) = bb$, $g(c) = ccc$ y $g(d) = d$ y definimos $L_3 = g^{-1}(L_2) = \{a^k d b^k d c^k \mid k \geq 0\}$. Por último, definimos el homomorfismo f como $f(a) = a$, $f(b) = b$, $f(c) = c$ y $f(d) = \lambda$ y definimos $L_4 = f(L_3) = \{a^k b^k c^k \mid k \geq 0\}$.

Dado que hemos partido del lenguaje L, le hemos aplicado sucesivas operaciones de cierre para la clase de los lenguajes incontextuales y hemos obtenido un lenguaje no incontextual la conclusión que obtenemos es que L no es incontextual.

2. Sean L_1 y L_2 dos lenguajes tales que $L_1 \cup L_2$ es recursivo y $L_1 \cap L_2$ también lo es. Demuestre que L_1 es recursivo si y sólo si lo es L_2.

(1 pto)

Solución

Demostraremos en primer lugar que si L_1 es recursivo entonces L_2 también lo es.

Observemos que L_2 lo podemos expresar como $L_2 = ((L_1 \cup L_2) - L_1) \cup (L_1 \cap L_2)$. Podemos aplicar la siguiente igualdad:

$$L_2 = ((L_1 \cup L_2) - L_1) \cup (L_1 \cap L_2) = ((L_1 \cup L_2) \cap \overline{L_1}) \cup (L_1 \cap L_2)$$

En la última expresión, debemos considerar que $L_1 \cup L_2$ es recursivo, que $\overline{L_1}$ es recursivo (ya que asumimos que L_1 lo es y la clase de los lenguajes recursivos es cerrada bajo complementario), que $L_1 \cap L_2$ es recursivo y que la clase de los lenguajes recursivos es cerrada bajo unión e intersección. Por lo tanto, al poder expresar L_2 como el resultado de aplicar operaciones de cierre de la clase de los lenguajes recursivos sobre lenguajes que lo son, L_2 es recursivo.

La demostración de que si L_2 es recursivo entonces L_1 también lo es se puede realizar de forma análoga a la anterior. Por lo tanto, hemos demostrado que L_1 es recursivo si y sólo si lo es L_2.

3. Sea $L_1 = \{a^i b^{2i} a^j \mid i, j \geq 0\}$ y sea $L_2 = \{a^i b^{2j} a^j \mid i, j \geq 0\}$. ¿ Son L_1, L_2 y $L_1 \cap L_2$ incontextuales ?

(2 ptos)

Solución

El lenguaje L_1 es incontextual ya que puede ser generado por la gramática incontextual definida por las siguientes reglas

$S \rightarrow AB$

$A \rightarrow aAbb \mid \lambda$

$B \rightarrow aB \mid \lambda$

El lenguaje L_2 también es incontextual ya que puede ser generado por la gramática incontextual definida por las siguientes reglas

$S \rightarrow AB$

$A \rightarrow aA \mid \lambda$

$B \rightarrow bbBa \mid \lambda$

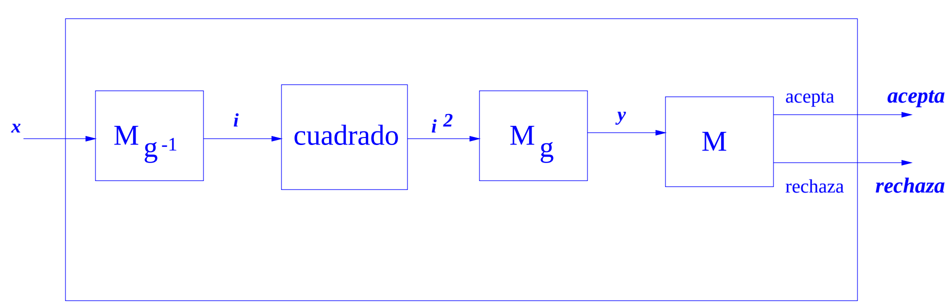
Por último, $L_1 \cap L_2 = \{a^i b^{2i} a^i \mid i \geq 0\}$ no es incontextual. Para demostrarlo, definiremos en primer lugar un homomorfismo h tal que $h(a) = a$, $h(b) = bb$ y $h(c) = a$. Obtenemos $h^{-1}(L_1 \cap L_2) = \{\{a,c\}^i b^i \{a,c\}^i \mid i \geq 0\}$. A continuación, obtenemos $h^{-1}(L_1 \cap L_2) \cap a^*b^*c^* = \{a^i b^i c^i \mid i \geq 0\}$ que, como se ha comprobado en numerosas ocasiones, no es incontextual. Al aplicarle a $L_1 \cap L_2$ operaciones de cierre para la clase de lenguajes incontextuales hemos obtenido un lenguaje no incontextual. Por lo tanto $L_1 \cap L_2$ no es incontextual.

4. Sea $g : \mathbb{N} \longrightarrow \Sigma^*$ una función que enumera Σ^* en orden canónico (lexicográfico). La función inversa $g^{-1} : \Sigma^* \longrightarrow \mathbb{N}$ asigna a una palabra su lugar en la enumeración. Dado un lenguaje L sobre Σ, se define $P(L) = \{x \in \Sigma^* \mid g((g^{-1}(x))^2) \in L\}$. Pronúnciese acerca del siguiente enunciado: "Si L es recursivo, entonces $P(L)$ también lo es".

(1 pto)

Solución

El enunciado es cierto. Partiremos de que L es recursivo y propondremos un esquema de aceptación recursivo para $P(L)$. En primer lugar, asumimos que g y g^{-1} son funciones computables totales y, en consecuencia, existen máquinas de Turing M_g y $M_{g^{-1}}$ que calculan respectivamente $g(i)$ con $i \in \mathbb{N}$ y $g^{-1}(x)$ con $x \in \Sigma^*$. Podemos contar también con una máquina de Turing M que acepta a L y siempre para, dado que partimos de que L es recursivo. De igual forma, podemos contar con una máquina de Turing que calcule el cuadrado de un número dado que esta función es computable total. A la máquina anterior la denotaremos por el módulo *cuadrado*. A partir de las anteriores máquinas proponemos el siguiente esquema de aceptación recursivo para $P(L)$



El funcionamiento de la anterior máquina se explica a continuación. Dada una cadena de entrada x, en primer lugar se calcula el valor $g^{-1}(x)$ mediante la máquina $M_{g^{-1}}$ dando como valor de salida i. A continuación se calcula el cuadrado de i mediante el módulo *cuadrado*. La salida del anterior módulo se toma como entrada para la máquina M_g que proporciona como salida $g(i^2) = g((g^{-1}(x))^2) = y$. Por último, se analiza si la anterior cadena y pertenece o no a L con la máquina M. En caso afirmativo se acepta la cadena de entrada y, en caso negativo, se rechaza. Dado que el anterior esquema garantiza siempre la parada y acepta $P(L)$, podemos concluir que $P(L)$ es recursivo.

5. Sea $\mathcal{A}$ el conjunto de máquinas de Turing que paran con todas las entradas y sea $\mathcal{B}$ el conjunto de máquinas de Turing que aceptan lenguajes recursivos. ¿ Qué relación existe entre ambos conjuntos?

(1 pto)

Solución

Se cumple que $\mathcal{A} \subset \mathcal{B}$. Tomemos, en primer lugar, una máquina M del conjunto $\mathcal{A}$. Por la definición que hemos enunciado, M para con cualquier entrada y, por lo tanto, el lenguaje que acepta es recursivo. Por lo tanto, M pertenece al conjunto $\mathcal{B}$ y, provisionalmente, $\mathcal{A} \subseteq \mathcal{B}$.

Veamos ahora que $\mathcal{A} \neq \mathcal{B}$. Tomemos, a modo de ejemplo, la máquina de Turing definida por la tupla $(\{q_0\}, \{a\}, \{a, B\}, \delta, q_0, B, \emptyset)$ donde la función δ se define como $\delta(q_0, a) = \delta(q_0, B) = (q_0, B, R)$.

Se puede comprobar fácilmente que la máquina definida anteriormente no para nunca ya que sus movimientos le obligan a seguir explorando nuevas celdas de cinta hacia la derecha, sea cual sea el símbolo que analice en cada momento. El lenguaje que acepta la anterior máquina es el lenguaje vacío (ya que no acepta ninguna cadena de entrada puesto que no para nunca). El lenguaje vacío es recursivo ya que se define trivialmente con una máquina de Turing que pare siempre rechazando la entrada.

Por lo tanto, la máquina que hemos definido pertenece al conjunto $\mathcal{B}$ (ya que su lenguaje es recursivo) y no pertenece al conjunto $\mathcal{A}$ (puesto que no para ante ninguna entrada). En consecuencia, $\mathcal{A} \neq \mathcal{B}$ y, a partir del primer razonamiento que hemos realizado, $\mathcal{A} \subset \mathcal{B}$.

(II) PROBLEMAS:

6. Escriba un módulo *Mathematica* que, tomando como entrada una lista, dé como salida otra lista formada por los elementos que en la lista de entrada ocupan posiciones pares.

(1 pto)

Solución

```
Solucion[entrada_List]:=Module[{ salida, k },
    salida = {};
    For[k=1, k ≤ Length[entrada], k=k+2,
        salida=Append[salida,entrada[[k]]]
    ];
    Return[salida]
]
```

7. Sea G la gramática definida por las reglas $S \to SA \mid BA \mid SB$; $A \to BS \mid b$; $B \to b$. Obtenga una gramática equivalente a G en Forma Normal de Greibach.

(1 pto)

Solución

En primer lugar, observemos que la gramática G del enunciado ya está totalmente simplificada ya que no contiene símbolos inútiles, producciones unitarias o producciones vacías. De igual forma, la gramática ya está en Forma Normal de Chomsky. Para aplicar el algoritmo de obtención de una gramática equivalente en Forma Normal de Greibach realizamos un cambio de auxiliares con $S = A_1$, $A = A_2$ y $B = A_3$. La gramática con los nuevos auxiliares es la siguiente

$A_1 \to A_1A_2 \mid A_3A_2 \mid A_1A_3$

$A_2 \to A_3A_1 \mid b$

$A_3 \to b$

A continuación, procedemos a eliminar la recursividad por la izquierda

$A_1 \to A_3A_2$

$B_1 \to A_2 \mid A_2B_1 \mid A_3 \mid A_3B_1$

$A_2 \to A_3A_1 \mid b$

$A_3 \to b$

Por último, procedemos a sustituir los primeros auxiliares que aparecen en las partes derechas de las producciones

$A_1 \to bA_2$

$B_1 \to bA_1 \mid b \mid bA_1B_1 \mid bB_1$

$A_2 \to bA_1 \mid b$

$A_3 \to b$

La anterior gramática ya está en Forma Normal de Greibach y es equivalente a la gramática del enunciado.

8. Sea G_1 definida por las reglas $S \to aSb \mid bSa \mid \lambda$. Sea G_2 definida por las reglas $S \to aA$; $A \to bS \mid b$. Sea el homomorfismo h definido como $h(a) = ab$ y $h(b) = b$. Obtenga una gramática que genere el lenguaje $(L(G_1))^*h(L(G_2))$.

(1 pto)

Solución

En primer lugar obtenemos una gramática para $(L(G_1))^*$

$S_1 \to SS_1 \mid \lambda$

$S \to aSb \mid bSa \mid \lambda$

A continuación, una gramática para $h(L(G_2))$

$S_h \to abA_h$

$A_h \to bS_h \mid b$

Por último, la gramática para $(L(G_1))^*h(L(G_2))$

$S_2 \to S_1S_h$

$S_1 \to SS_1 \mid \lambda$

$S \to aSb \mid bSa \mid \lambda$

$S_h \to abA_h$

$A_h \to bS_h \mid b$