sol95a.pdf

Anónimo

Computabilidad y complejidad

3º Grado en Ingeniería Informática

Escuela Técnica Superior de Ingeniería Informática
Universidad Politécnica de Valencia

Examen de Computabilidad y Complejidad

(CMC)

7 de julio de 1995

(I) CUESTIONES (justifique formalmente las respuestas)

1. Sea $\mathcal{L}$ la clase de los lenguajes recursivamente enumerables que no son recursivos. ¿Es $\mathcal{L}$ cerrada bajo complementación ?

   (1 punto)

   Solución

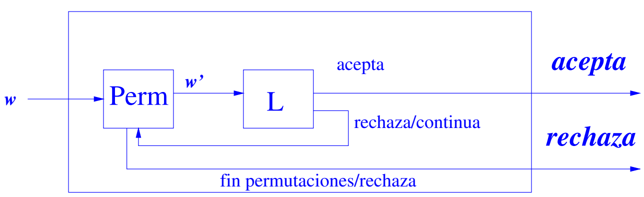
   La clase $\mathcal{L}$ no es cerrada bajo complementación. Por contradicción, supongamos que L y $\overline{L}$ son recursivamente enumerables, entonces, tal y como se ha visto en clase, L y $\overline{L}$ son recursivos y, por lo tanto, no pertenecen a la clase enunciada, lo cual es contradictorio.

2. Sea $L \subseteq \Sigma^*$ un lenguaje recursivo. ¿ Es recursivo el lenguaje definido como $L' = \{v \in \Sigma^* \mid \exists w \in L \wedge v$ es una permutación de $w\}$.

   (1 punto)

   Solución

   L' es recursivo. Actuaremos construyendo una máquina de Turing que acepte L' y que pare ante cualquier entrada. Para ello, partiremos de dos módulos: L que es una máquina de Turing que acepta a L y siempre para (obsérvese que L es recursivo) y $Perm$ que calcula todas las permutaciones de una cadena de entrada. Obviamente, el módulo $Perm$ obedece a un algoritmo que finaliza en tiempo finito. Con los dos anteriores módulos, construimos la máquina que se muestra a continuación



   El funcionamiento de la máquina es como sigue: Inicialmente se pasa la cadena de entrada w al módulo $Perm$ que calcula las permutaciones de la cadena. De forma iterativa, se le pasa cada permutación calculada w' al módulo L que establece si la permutación pertenece o no al lenguaje L. En caso de aceptación, se acepta la cadena w ya que se cumplen los requisitos del lenguaje. Si ninguna permutación pertenece a L entonces se rechaza la cadena de entrada y la máquina para. La condición de parada queda garantizada por las limitaciones temporales de los anteriores módulos.

3. Sea la operación $\mathcal{P}$ definida sobre cadenas como $\mathcal{P}(x) = xx^{-1}$. Se extiende la operación a lenguajes como $\mathcal{P}(L) = \{\mathcal{P}(x) \mid x \in L\}$. ¿ Es la familia de los lenguajes incontextuales cerrada bajo la operación $\mathcal{P}$?. (x^{-1} denota el reverso o inverso de la cadena x).

(1.5 puntos)

Solución

La operación $\mathcal{P}$ no es de cierre. Para ello, tomemos el lenguaje $L = \{a^n b^n : n \geq 1\}$ que es incontextual al ser generado por la gramática $S \to aSb \mid ab$. Al aplicar la operación $\mathcal{P}$ al lenguaje L obtenemos el lenguaje $L' = \{a^n b^n b^n a^n : n \geq 1\}$. El lenguaje L' no es incontextual. Esto lo podemos demostrar por propiedades de cierre: Definamos en primer lugar el homomorfismo h de forma que $h(a) = a$, $h(b) = bb$ y $h(c) = a$. Entonces, $h^{-1}(L') = \{\{a,c\}^n b^n \{a,c\}^n : n \geq 1\}$. Haciendo la intersección del anterior lenguaje con la expresión regular $aa^*bb^*cc^*$ obtenemos el lenguaje $\{a^n b^n c^n : n \geq 1\}$ que, como se ha visto en repetidas ocasiones, no es incontextual. Por lo tanto, $\mathcal{P}$ no es de cierre.

4. Sea $L = \bigcup_{n \geq 0} (1^n 0^n)^*$. ¿ Es L un lenguaje incontextual ?.

(1 punto)

Solución

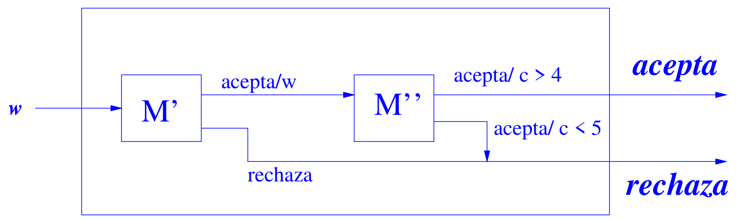
L no es incontextual. Lo demostraremos por propiedades de cierre. Tomemos L y hagamos la intersección con la expresión regular $11^*00^*11^*00^*$. Obtenemos el lenguaje $L' = \{1^n 0^n 1^n 0^n : n \geq 1\}$. Podemos demostrar que L' no es incontextual: Definamos el homomorfismo h de forma que $h(a) = 1$, $h(b) = 0$, $h(c) = 1$ y $h(d) = 0$. Aplicamos h^{-1} a L' y obtenemos $\{\{a,c\}^n \{b,d\}^n \{a,c\}^n \{b,d\}^n : n \geq 1\}$. Haciendo la intersección con la expresión regular $aa^*bb^*cc^*dd^*$ obtenemos el lenguaje $\{a^n b^n c^n d^n : n \geq 1\}$. Por último, definimos el homomorfismo g como $g(a) = a$, $g(b) = b$, $g(c) = c$ y $g(d) = \lambda$ y, aplicándolo sobre el último lenguaje, obtenemos $\{a^n b^n c^n : n \geq 1\}$ que, como se ha visto, no es incontextual. Por lo tanto, si al aplicarle a L propiedades de cierre obtenemos un lenguaje no incontextual, la conclusión a la que llegamos es que L tampoco lo es.

5. Sea M una máquina de Turing y se define $L_5(M) = \{x \in \Sigma^* \mid x \in L(M) \wedge M$ pasa más de 5 veces por el estado inicial al computar $x\}$. Pronúnciese sobre la veracidad o falsedad de la siguiente afirmación: *Si $L(M)$ es recursivo entonces $L_5(M)$ también lo es.*

(1.5 puntos)

Solución

La afirmación es cierta. Definamos M' como una máquina de Turing que acepta $L(M)$ y para ante cualquier entrada (obsérvese que $L(M)$ es recursivo). Podemos construir una máquina M'' que simule a M y que lleve un contador c inicializado a cero y que se incremente cada vez que M pasa desde un estado cualquiera al estado inicial. A partir de M' y M'' podemos construir la siguiente máquina:



La anterior máquina funciona como sigue: Inicialmente se establece si la cadena de entrada w pertenece o no a $L(M)$ (esto lo hacemos con M'). Si la cadena es rechazada entonces rechazamos la entrada. Si la cadena w pertenece a $L(M)$ entonces se la pasamos a M'' que establece cuantas veces pasa M por su estado inicial. Si el contador c supera el valor 4 entonces aceptamos la entrada, en caso contrario la rechazamos. La condición de parada queda garantizada en cualquier caso. Por lo tanto $L_5(M)$ es recursivo.

(II) PROBLEMAS:

6. Dada la gramática incontextual G definida por las producciones

$$S \to SaS \mid bS \mid \lambda$$

y la sustitución $f : \{a,b\} \to \mathcal{P}(\{a,b\}^*)$ con $f(a) = (L(G))^*$ y $f(b) = (L(G))(L(G) - \{\lambda\})$. Obtener una gramática incontextual que genere $L(G) \cup f(L(G))$.

(2 puntos)

Solución

En primer lugar obtendremos las gramáticas para la sustitución f. Para $f(a)$ definimos la gramática G_a mediante las producciones

$S_a \to S'S_a \mid \lambda$

$S' \to S'aS' \mid bS' \mid \lambda$

Para $f(b)$ definimos la gramática G_b mediante las producciones

$S_b \to S''S'''$

$S'' \to S''aS'' \mid bS'' \mid \lambda$

$S''' \to S'''aS''' \mid aS''' \mid S'''a \mid a \mid bS''' \mid b$

La gramática para $f(L(G))$ es la siguiente

$S_f \to S_fS_aS_f \mid S_bS_f \mid \lambda$

$S_a \to S'S_a \mid \lambda$

$S' \to S'aS' \mid bS' \mid \lambda$

$S_b \to S''S'''$

$S'' \to S''aS'' \mid bS'' \mid \lambda$

$S''' \to S'''aS''' \mid aS''' \mid S'''a \mid a \mid bS''' \mid b$

Por último, la gramática para $L(G) \cup f(L(G))$ se define mediante las producciones

$S_\cup \to S \mid S_f$

$S \to SaS \mid bS \mid \lambda$

$S_f \to S_fS_aS_f \mid S_bS_f \mid \lambda$

$S_a \to S'S_a \mid \lambda$

$S' \to S'aS' \mid bS' \mid \lambda$

$S_b \to S''S'''$

$S'' \to S''aS'' \mid bS'' \mid \lambda$

$S''' \to S'''aS''' \mid aS''' \mid S'''a \mid a \mid bS''' \mid b$

7. Dada la gramática G:

$$\begin{array}{ll} S \to Sa \mid ASB \mid a \mid \lambda & B \to ABA \mid a \\ A \to Ab \mid Aa \mid b \mid C & C \to Ca \mid Cb \mid D \\ D \to aDa \mid DC & E \to aEb \mid \lambda \end{array}$$

Obtener una gramática incontextual G' en Forma Normal de Greibach de modo que $L(G') = (L(G) - \{\lambda\})$.

(2 puntos)

Solución

En primer lugar, procederemos a simplificar la gramática G

Símbolos no generativos: $\{C, D\}$

Gramática sin símbolos no generativos:

$$\begin{array}{ll} S \to Sa \mid ASB \mid a \mid \lambda & B \to ABA \mid a \\ A \to Ab \mid Aa \mid b & E \to aEb \mid \lambda \end{array}$$

Símbolos no alcanzables: $\{E\}$

Gramática sin símbolos no alcanzables:

$$\begin{array}{l} S \to Sa \mid ASB \mid a \mid \lambda \\ A \to Ab \mid Aa \mid b \\ B \to ABA \mid a \end{array}$$

Símbolos anulables: $\{S\}$

Gramática sin producciones vacías:

$$\begin{array}{l} S \to Sa \mid a \mid ASB \mid AB \\ A \to Ab \mid Aa \mid b \\ B \to ABA \mid a \end{array}$$

La anterior gramática ya está totalmente simplificada ya que se puede comprobar que no contiene ni producciones unitarias no símbolos inútiles.

Procedemos ahora a transformar la anterior gramática de forma que no hayan auxiliares y terminales juntos o más de un terminal en cada parte derecha

$$\begin{array}{l} S \to SS_a \mid a \mid ASB \mid AB \\ A \to AS_b \mid AS_a \mid b \\ B \to ABA \mid a \\ S_a \to a \\ S_b \to b \end{array}$$

Pasamos ahora a obtener la gramática en Forma Normal de Greibach eliminando en primer lugar las recursividades por la izquierda que aparecen en las producciones

$$\begin{array}{l} S \to a \mid ASB \mid AB \mid aS' \mid ASBS' \mid ABS' \\ S' \to S_a \mid S_aS' \\ A \to b \mid bA' \\ A' \to S_b \mid S_bA' \mid S_a \mid S_aA' \\ B \to ABA \mid a \\ S_a \to a \\ S_b \to b \end{array}$$

Por último, hacemos las transformaciones finales y obtenemos la gramática en Forma Normal de Greibach

$S \to a \mid bSB \mid bA'SB \mid bB \mid bA'B \mid aS' \mid bSBS' \mid bA'SBS' \mid bBS' \mid bA'BS'$
$S' \to a \mid aS'$
$A \to b \mid bA'$
$A' \to b \mid bA' \mid a \mid aA'$
$B \to bBA \mid bA'BA \mid a$