sol99b.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)

16 de septiembre de 1999

(I) Cuestiones (justifique formalmente las respuestas)

1. Sea el lenguaje $L = \{x\#y : x, y \in (0+1)^*, x = y \vee x \text{ no contiene ningún } 0\}$. ¿ Es L incontextual ?

(1.5 ptos)

Solución

El lenguaje L no es incontextual. Lo demostraremos mediante el lema de bombeo. Tomemos n como la constante del lema y la cadena $z = 0^n1^n\#0^n1^n$ que pertenece a L y cumple las condiciones de partida del lema.

Factorizamos, de la forma habitual, $z = uvwxy$ y veremos las distintas posibilidades de localización de las subcadenas v y x. Consideraremos al primer bloque de la cadena como la subcadena anterior al símbolo $\#$, mientras que el segundo bloque será la subcadena posterior al símbolo $\#$.

(a) vx formado por símbolos 0 del primer bloque con $|vx| = k \geq 1$. En este caso tomamos el valor $i = 2$ y formamos la cadena $uv^2wx^2y = 0^{n+k}1^n\#0^n1^n$ que no pertenece a L ya que el primer bloque no coincide con el segundo bloque y, además, el primer bloque sigue conteniendo símbolos 0.

(b) vx formado por símbolos 1 del primer bloque con $|vx| = k \geq 1$. En este caso tomamos el valor $i = 0$ y formamos la cadena $uv^2wx^2y = 0^n1^{n-k}\#0^n1^n$ que no pertenece a L ya que el primer bloque no coincide con el segundo bloque y, además, el primer bloque sigue conteniendo símbolos 0.

(c) vx formado por símbolos 0 del segundo bloque con $|vx| = k \geq 1$. El análisis de este caso es idéntico al del caso (a).

(d) vx formado por símbolos 1 del segundo bloque con $|vx| = k \geq 1$. El análisis de este caso es idéntico al del caso (b).

(e) vx formado por símbolos 0 y 1 del primer bloque, con $|vx|_0 = k$ y $|vx|_1 = j$ donde $k, j \geq 1$. Tomamos el valor $i = 2$ y se forma la cadena $z_1\#0^n1^n$ que no pertenece a L ya que el primer bloque no coincide con el segundo bloque, puesto que $|z_1|_0 = n + k$ y $|z_1|_1 = n + j$ y, además, el primer bloque sigue conteniendo símbolos 0.

(f) vx formado por símbolos 0 y 1 del segundo bloque, con $|vx|_0 = k$ y $|vx|_1 = j$ donde $k, j \geq 1$. El análisis de este caso coincide con el del caso anterior.

(g) vx formado por símbolos 1 del primer bloque y símbolos 0 del segundo bloque, con $|vx|_1 = k$ y $|vx|_0 = j$ donde $k, j \geq 1$. Tomamos el valor $i = 2$ y se forma la cadena $z_1\#z_2$ que no pertenece a L ya que el primer bloque no coincide con el segundo bloque, puesto que $|z|_1 = n+k$, $|z_2|_1 = n$, $|z_1|_0 = n$ y $|z_2|_0 = n+j$ y, además, el primer bloque sigue conteniendo símbolos 0.

(h) vx contiene el símbolo $\#$. En este caso tomando el valor $i = 2$ y se forma la cadena $uvvwxxy$ que contiene más de un símbolo $\#$ y, por lo tanto, no pertenece a L.

Debido a las condiciones 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 L no es incontextual.

2. Sea el lenguaje $L \subset \{0,1\}^*$ formado por todas las palabras de longitud impar tales que el primer símbolo, el símbolo central y el último símbolo coinciden. ¿ Es L incontextual ?

(1.5 ptos)

Solución

El lenguaje del enunciado, en contra de lo que pudiera parecer, sí es incontextual. La siguiente gramática incontextual, donde S es el axioma, genera L

$S \rightarrow 0A0 \mid 1B1 \mid 0 \mid 1$

$A \rightarrow 0A0 \mid 0A1 \mid 1A0 \mid 1A1 \mid 0$

$B \rightarrow 0B0 \mid 0B1 \mid 1B0 \mid 1B1 \mid 1$

La anterior gramática sólo genera cadenas impares. Si el símbolo inicial y final es 0, entonces se utiliza la regla $S \rightarrow 0A0$ y, posteriormente, a partir de A se finaliza con el símbolo central 0. Si el símbolo inicial y final es 1, entonces se utiliza la regla $S \rightarrow 1B1$ y, posteriormente, a partir de B se finaliza con el símbolo central 1. Finalmente, las reglas $S \rightarrow 0 \mid 1$ se corresponden con los casos triviales de cadenas con un único símbolo.

3. Se define f de forma que, a partir de tres lenguajes L_1, L_2 y L_3, $f(L_1, L_2, L_3)$ es el lenguaje formado por las palabras que pertenecen al menos a dos de ellos. ¿ Es la familia de los lenguajes recursivos cerrada bajo f ?

(1.5 ptos)

Solución

La clase de los lenguajes recursivos es cerrada bajo f. En primer lugar, debemos partir de que la clase de los lenguajes recursivos es cerrada bajo unión e intersección. La operación f la podemos expresar como $f(L_1, L_2, L_3) = (L_1 \cap L_2) \cup (L_1 \cap L_3) \cup (L_2 \cap L_3)$. De esta forma, puesto que f se puede representar como el resultado de aplicar un número finito de veces operaciones de intersección y unión, podemos concluir que la familia de los lenguajes recursivos es cerrada bajo f.

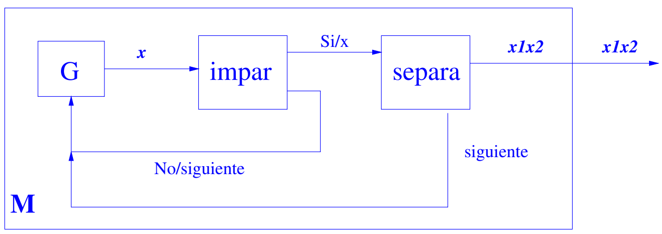
4. Sea $L \subseteq \Sigma^*$. Se define la operación P de forma que $P(L) = \{x_1x_2 : |x_1| = |x_2| \wedge \exists a \in \Sigma : x_1ax_2 \in L\}$. ¿Es la clase de los lenguajes recursivamente enumerables cerrada bajo P ? (Ejemplo: $L = \{abab, bba\}, P(L) = \{ba\}$)

(1.5 ptos)

Solución

La clase de los lenguajes recursivamente enumerables es cerrada bajo la operación P. Obsérvese que la operación P se limita a seleccionar las cadenas impares del lenguaje L y elimina su símbolo central. Para demostrar que $P(L)$ es recursivamente enumerable, partiendo de que L lo es, construiremos una máguina de Turing M que genere el lenguaje $P(L)$. Para ello contaremos con un módulo G que genera el

lenguaje L. Contaremos también con un módulo *impar* que detecta si una cadena de entrada tiene longitud impar y con un módulo *separa* que, dada una cadena impar de entrada, elimina su símbolo central. La existencia del módulo G se basa en el hecho de que L es recursivamente enumerable, mientras que los módulos *impar* y *separa* se fundamentan en máquinas de Turing multicintas con algoritmos de funcionamiento triviales. A partir de los anteriores módulos, el esquema de la máquina M que se propone es el siguiente



El funcionamiento de M se explica a continuación. El módulo G genera las cadenas de L. Cada cadena generada por G se pasa como entrada al módulo *impar*. Si la cadena tiene longitud impar, entonces se pasa como entrada al módulo *separa* que elimina su símbolo central, la emite como salida y vuelve a activar el módulo G. Si la cadena es de longitud par, entonces el módulo *impar* no emite ninguna entrada para el módulo *separa* y activa de nuevo el módulo G. De esta forma, M genera todas las cadenas de $P(L)$ y, por lo tanto, $P(L)$ es recursivamente enumerable por lo que P es una operación de cierre para la clase de lenguajes recursivamente enumerables.

(II) PROBLEMAS:

5. Dada una gramática incontextual, diremos que un símbolo auxiliar A es *recursivo* si la gramática contiene alguna producción con la forma $A \to \alpha A \beta$. Se pide escribir un módulo *Mathematica* que, dada una gramática incontextual como parámetro, devuelva *True* o *False* en función de que la gramática contenga algún símbolo auxiliar *recursivo*.

(2 ptos)

Solución

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

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

$S \to AS \mid CCA \mid SEF \mid AB$
$A \to Ba \mid Aa$
$B \to bS \mid CbD \mid BB \mid \lambda$
$C \to DEc \mid EFc \mid CC$
$D \to CDF \mid ddF \mid FE$
$E \to eEa \mid Eea \mid eaC$
$F \to ES \mid FE \mid d$

(2 ptos)

Solución

Procedemos, en primer lugar, a simplificar la gramática G.

Eliminación de símbolos no generativos

Símbolos no generativos: $\{C, E\}$

Gramática sin símbolos no generativos

$S \to AS \mid AB$
$A \to Ba \mid Aa$
$B \to bS \mid BB \mid \lambda$
$D \to ddF$
$F \to d$

Eliminación de símbolos no alcanzables

Símbolos no alcanzables: $\{D, F\}$

Gramática sin símbolos no alcanzables

$S \to AS \mid AB$
$A \to Ba \mid Aa$
$B \to bS \mid BB \mid \lambda$

Eliminación de producciones vacías

Símbolos anulables: $\{B\}$

Gramática sin producciones vacías

$S \to AS \mid AB \mid A$
$A \to Ba \mid a \mid Aa$
$B \to bS \mid BB \mid B$

Eliminación de producciones unitarias

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

Gramática sin producciones unitarias

$S \to AS \mid AB \mid Ba \mid a \mid Aa$
$A \to Ba \mid a \mid Aa$
$B \to bS \mid BB$

La anterior gramática 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

$S \rightarrow AS \mid AB \mid BC_a \mid a \mid AC_a$

$A \rightarrow BC_a \mid a \mid AC_a$

$B \rightarrow C_bS \mid BB$

$C_a \rightarrow a$

$C_b \rightarrow b$

Obsérvese que la anterior gramática ya está en Forma Normal de Chomsky y, por lo tanto, no es necesario realizar una factorización en sus producciones.