

Teoría Matemática de la Computación

Segundo Problemario

Prof. Miguel A. Pizaña
16 de Junio de 2006

I Lenguajes Regulares

1. Probar que los lenguajes regulares son cerrados bajo: complemento, unión, intersección, concatenación y cerradura de Kleene.
2. Proporcione un algoritmo (en pseudocódigo) que para cualquier AFD $M = (Q, \Sigma, \delta, q_0, F)$ dado, decida si:
 - (a) $L(M)$ es vacío o no.
 - (b) $L(M)$ es finito o infinito.
 - (c) $L(M)$ es cofinito o no.
3. Realizar las transformaciones indicadas usando los algoritmos vistos en clase.
 - (a) Transformar los AFN- ε 's de la Figura 1 en AFN's
 - (b) Transformar los AFN's de la Figura 2 en AFD's
 - (c) Transformar los AFD's de la Figura 3 en expresiones regulares.
 - (d) Transformar las siguientes expresiones regulares en AFN- ε 's
 1. $(0 + 1)^*00(0 + 1)^*$
 2. $(0 + \varepsilon)(1 + 10)^*$
 3. $(0 + 1)^*0(0 + 1)^7$
4. Determinar si los siguientes lenguajes son regulares o no (Encuentre un AFD que lo reconozca o use el lema del bombeo).
 - (a) $\{\omega \in \{a, b, c, d\}^* : \omega \text{ contiene a la palabra } abcd \text{ como subcadena} \}$
 - (b) $\{\omega \in \{a, b, c, d\}^* : \omega \text{ NO contiene a la palabra } abcd \text{ como subcadena} \}$
 - (c) $L_m = \{0^{n+m}1^n : n \geq 0\} \cup \{0^n1^{n+m} : n \geq 0\}$

- (d) $\bigcup_{m=0}^{\infty} L_m$
(Aquí L_m es como en el problema anterior)
- (e) $\{0^{n^2} : n \geq 1\}$
- (f) {Sonetos en español}
(suponga que el conjunto de cadenas del “español” está bien definido)
- (g) {Paréntesis balanceados} =
= $\{(), (()), ()(), ((())), (()()), ()()(), (((()))), (((()()))), \dots, (((()))(), \dots\}$
- (h) Cadenas de paréntesis balanceados que no tengan más de un trillón de paréntesis.
- (i) {Expresiones aritméticas} =
= $\{\omega \in \{+, -, *, /, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}^* : \text{etcétera}\}$
(considerar a “-” y “+” únicamente como operadores binarios).
- (j) {Expresiones aritméticas con paréntesis} =
= $\{\omega \in \{+, -, *, /, (,), 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}^* : \text{etcétera}\}$
- (k) {Declaraciones de variables en C}
- (l) {Programas en C}

II Lenguajes recursivos y recursivamente enumerables

- ¿Qué es una máquina de Turing? ¿Algoritmo? ¿Lenguaje recursivamente enumerable? ¿Lenguaje recursivo? Dada una máquina de Turing M, ¿Cómo se define el lenguaje aceptado por M?
- Para cada uno de los siguientes lenguajes proporcione un algoritmo que los reconozca:
 - $\{0^n 1^n : n \in \mathbb{N}\}$.
 - { paréntesis balanceados }.
 - { expresiones aritméticas con paréntesis }
 - { palíndromos de 0's y 1's }.
 - $L_m = \{0^{n+m} 1^n : n \geq 0\} \cup \{0^n 1^{n+m} : n \geq 0\}$.
 - $\{w \in \{0, 1\}^* : w \text{ tiene igual número de 0's que de 1's}\}$.
- Describa cinco variaciones equivalentes de la máquina de Turing básica.
- Demuestre que todo lenguaje regular es recursivo.

5. Demuestre que todo lenguaje recursivo es recursivamente enumerable.
6. Demuestre que existen lenguajes que son:
 - (a) No recursivamente enumerables.
 - (b) Recursivamente enumerables pero no recursivos.
 - (c) Recursivos pero no regulares.
7. Enuncie la tesis de Church-Turing. La tesis de Church-Turing ¿puede ser demostrada matemáticamente? Explique.
8. Enuncie el problema del paro.
9. ¿Qué es un problema irresoluble?
10. Describa tres problemas irresolubles.
11. Enuncie el teorema de Rice.
12. ¿Algún día se podrá hacer un antivirus infalible?
13. ¿Puede el cerebro humano hacer algo que las Máquinas de Turing no?
14. ¿Pueden las máquinas pensar?

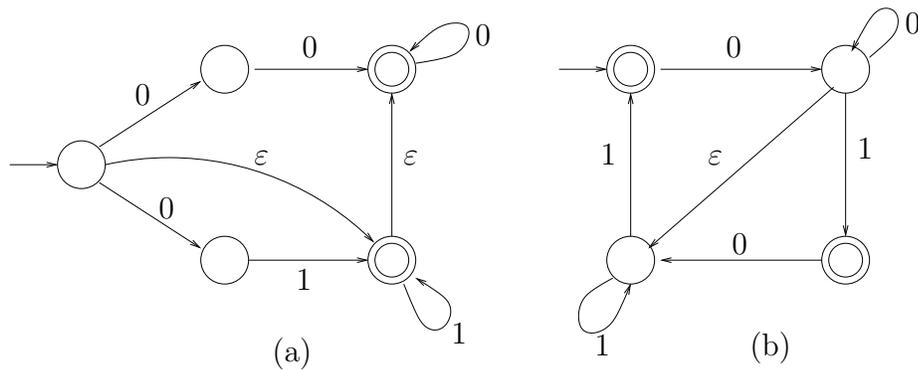


Figura 1: Convertir de AFN- ϵ a AFN

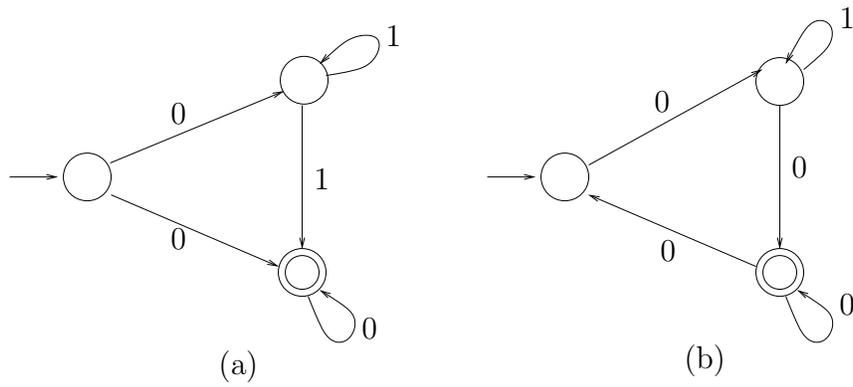


Figura 2: Convertir de AFN a AFD

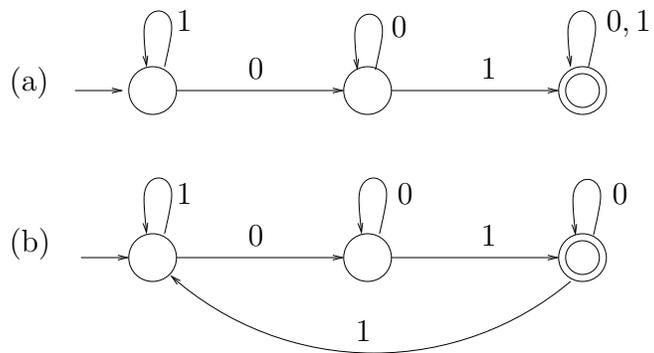


Figura 3: Convertir de AFD a expresión regular