Teoría Matemática de la Computación
Segundo Problemario

Prof. Miguel A. Pizaña

29 de marzo de 2011

I Máquinas de Turing.

1. ¿Qué es un a Máquina de Turing? ¿Cómo se define? ¿Cómo se llaman las teorías que emergen del estudio de este modelo? ¿Cuáles son sus principales características? ¿para que sirve? ¿Cuánta memoria tiene? ¿Qué operaciones básicas sabe realizar una MT? ¿Cuáles aspectos de la realidad aproxima bien este modelo? ¿Cuáles aproxima mal? ¿Cuántos y cuáles son los supuestos falsos en los que se basa este modelo?
2. Leer *On Computable Numbers, with an Applicatrion to the Entscheidungsproblem* de Alan Turing. Luego responda: ¿Cómo le llama Turing a sus máquinas? ¿Qué diferencias hay entre su definición y la definición vista en clase? ¿Qué es para Turing un número computable? ¿son computables $\pi, \sqrt{2}, \frac{1}{3}, \tan(23)$? ¿Cuál es la razón por la que la máquina usa una cinta para hacer anotaciones? ¿Por qué usa un número finito de símbolos? ¿Qué es el Entscheidungsproblem? ¿Qué prueba Turing al respecto?
3. Leer *Computer Machinery and Intelligence* de Alan Turing. Luego conteste: ¿Qué es la prueba de Turing (Turing's Test)? ¿Usted cree que alguna máquina podrá algún día aprobar la prueba de Turing? Suponiendo que sí, ¿Eso indicaría que la máquina tiene inteligencia? ¿Puede el cerebro humano hacer algo que las Máquinas de Turing no? Según Turing, ¿pueden las máquinas pensar? ¿Qué argumentos en favor y en contra da Turing? ¿Qué dice la objeción delcuarto chino? ¿Qué dice la contra-objeción del tomografía cerebral a escala molecular? Según usted, ¿pueden las máquinas pensar? ¿por qué?

4. ¿Qué dice la tesis de Church-Turing? ¿Cree usted en la tesis de Church-Turing? ¿Puede esta tesis ser demostrada matemáticamente? Explique.
5. ¿Qué tan veloz es una Máquina de Turing? ¿cómo se puede comparar su velocidad con la velocidad de la computadora que tiene en su casa? ¿qué dice el teorema de aceleración lineal? ¿cuál es la idea de la prueba de ello? ¿se ha usado este teorema para mejorar el rendimientode las computadoras reales?
6. ¿Hay algo que la computadora de tu casa puede hacer pero que no puede hacer una Máquina de Turing? ¿Hay algo que la máquina de Turing puede hacer, pero la computadora de tu casa no?
7. En cada inciso proporcione una Máquina de Turing que:
   (a) Inserte un símbolo al inicio de la cinta.
   (b) Dado un número binario no negativo n, calcule $n + 1$.
   (c) Dado un número binario positivo n, calcule $n - 1$.
   (d) Dado un par de número binarios separados por un "#", calcule la suma.
8. Para cada uno de los siguientes lenguajes proporcione una máquina de Turing (que siempre para) que los reconozca:
   (a) $\{n : n$ es par $\}$
   (b) $\{0^n 1^n : n \in \mathbb{N}\}$.
   (c) $\{$ paréntesis balanceados $\}$.
   (d) $\{$ expresiones aritméticas con paréntesis $\}$
   (e) $\{$ palíndromos de 0's y 1's $\}$.
   (f) $L_m = \{0^{n+m}1^n : n \geq 0\} \cup \{0^n 1^{n+m} : n \geq 0\}$.
   (g) $\{w \in \{0,1\}^* : w$ tiene igual número de 0's que de 1's $\}$.
9. Demuestre que todo lo que un AFD puede hacer, también lo puede hacer alguna máquina de Turing.
10. ¿Qué significa que un problema sea irresoluble?
11. ¿Qué dice el teorema de Rice? ¿Qué consecuencias tiene esto en la práctica?
12. ¿Qué es una Máquina de Turing No Determinista? Si P =polinomial, ¿que es NP? ¿Es cierto que P = NP? ¿Es cierto que $\mathrm{P} \cap \mathrm{NP} = \varnothing$?

II Autómatas finitos Deterministas.

1. ¿Qué es un Autómata Finito Determinista? ¿Cómo se define? ¿Cuáles son sus principales características? ¿Para qué sirve? ¿Cómo se define el *lenguaje reconocido* por un AFD? ¿Cuáles aspectos de la realidad aproxima bien este modelo? ¿Cuáles aproxima mal?
2. Para cada uno de los siguientes lenguajes, proporcione un AFD que los reconozca (suponga que $\Sigma = \{0, 1\}$).
   (a) Cadenas que terminen en 00.
   (b) Cadenas que contengan tres (o más) ceros consecutivos.
   (c) Cadenas que contengan tres (pero no más de tres) ceros consecutivos.
   (d) Cadenas que contengan la subcadena 000 exactamente una vez.
   (e) Cadenas que representen números enteros divisibles entre 3.
   (f) Cadenas que representen números enteros divisibles entre 7.
   (g) Cadenas que tengan un 0 en la décima posición contando desde la izquierda.
   (h) Lo mismo que el anterior, pero contando desde la derecha.
   (i) Cadenas que contengan a 0101 como subcadena.
   (j) Cadenas que NO contengan a 0101 como subcadena.
   (k) Cadenas tal que en cada subcadena de longitud 5 haya al menos 2 ceros.
   (l) Cadenas tal que en cada prefijo el valor absoluto de la diferencia entre el número de 1's y el número de 0's sea menor o igual a 5.
   (m) Cadenas como en el inciso anterior en donde además el número total de ceros es igual al número total de unos.
3. Haga un lista completa de todos los AFD con alfabeto binario que tengan uno o dos estados (son 66, pero muchos de ellos son equivalentes). ¿Qué lenguaje reconoce cada uno de ellos?
4. Demuestre que si permitimos que el conjunto de estados en un Autómata Determinista sea infinito, entonces, dado cualquier lenguaje $L \subseteq \Sigma^*$ existe un Autómata Determinista (posiblemente infinito) M tal que $L = L(M)$.

5. Demuestre que dado cualquier lenguaje **finito** L existe un AFD que lo reconoce.

6. ¿Existe un AFD que dada cualquier matriz de $n \times n$ ($n < 50$) de números de punto flotante de doble precisión (todo ello codificado como una cadena binaria) decida si es invertible? Explique.

7. Los AFD sólo modelan algoritmos de decisión. Desarrolle usted su propio modelo matemático basado en el del AFD (memoria finita) que modele **cualquier** procesamiento de información $f : \Sigma^* \to \Sigma^*$ "computable con memoria finita". Por ejemplo, su modelo debería poder transformar cualquier cadena $w \in \Sigma^*$ de ceros y unos en la cadena w' que consta del doble de unos que w y no tiene ceros.

8. Modifique su modelo anterior para que sea capaz de percibir el paso del tiempo y actuar en respuesta a ello. Por ejemplo, su modelo debería poder mostrar el comportamiento del teclado: Si oprimes y sueltas (rápidamente) una tecla aparece la correspondiente letra en pantalla, pero si oprimes y mantienes oprimida la tecla aparecerá la tecla correspondiente con un número de repeticiones que depende del tiempo transcurrido.

9. ¿Puede usted hacer un programa de computadora (con memoria finita) que lea un número natural n expresado en binario y regrese $n+1$ también expresado en binario? ¿Y si el número debe ser leído y escrito al revés (es decir en formato *bigendian*)? ¿Puede hacer un programa que regrese $2n$? ¿$2n + 1$? ¿n^2?.

10. Un lenguaje L es *cofinito* si su complemento $\bar{L} = \Sigma^* - L$ es finito. Demuestre que todo lenguaje cofinito es regular.

11. Demuestre, para cualquier autómata M, con función de transición δ se tiene que $\delta(q, uv) = \delta(\delta(q, u), v)$.

12. Demuestre el lema siguiente. Muestre que el lema también vale si el conjunto cadenas $X := \{x_1, x_2, \ldots\}$ es infinito.

    Lema Si L es un lenguaje regular y suponga que existen cadenas $x_1, x_2, \ldots, x_n, z_{12}, z_{13}, z_{23}, \ldots, z_{ij}, \ldots z_{(n-1)n} \in \Sigma^*$ que satisfacen: ($x_i z_{ij} \in L$ y $x_j z_{ij} \notin L$) ó ($x_i z_{ij} \notin L$ y $x_j z_{ij} \in L$). Entonces, cualquier AFD que reconozca a L debe tener almenos n estados.

13. Muestre que cualquier AFD que reconozca a $L = (0 + 1)^* 0 (0 + 1)^n$ necesita al menos 2^{n+1} estados.

14. Enuncie y demuestre el Lema alfa.

15. Determinar si los siguientes lenguajes son regulares o no (Encuentre un AFD que lo reconozca o use el lema alfa).
    (a) $\{\omega \in \{a,b,c,d\}^* : \omega$ contiene a la palabra *abcd* como subcadena $\}$
    (b) $\{\omega \in \{a,b,c,d\}^* : \omega$ **NO** contiene a la palabra *abcd* como subcadena $\}$
    (c) $L_m = \{0^{n+m}1^n : n \geq 0\} \cup \{0^n 1^{n+m} : n \geq 0\}$
    (d) $\bigcup_{m=0}^{\infty} L_m$
        (Aquí L_m es como en el problema anterior)
    (e) $\{0^{n^2} : n \geq 1\}$
    (f) $\{0^p : p$ es primo $\}$.
    (g) {Sonetos en español}
        (suponga que el conjunto de cadenas del "español" está bien definido)
    (h) {Paréntesis balanceados} =
        $= \{(), (()), ()(), ((())), (())(), ()(()), ()()(), ((()))), ((()())), \ldots, ((()))(), \ldots\}$
    (i) Cadenas de paréntesis balanceados que no tengan más de un trillón de paréntesis.
    (j) {Expresiones aritméticas} =
        $= \{\omega \in \{+,-,*,/,0,1,2,3,4,5,6,7,8,9\}^* :$ etcétera $\}$
        (considerar a "$-$" y "$+$" únicamente como operadores binarios).
    (k) {Expresiones aritméticas con paréntesis} =
        $= \{\omega \in \{+,-,*,/,(,),0,1,2,3,4,5,6,7,8,9\}^* :$ etcétera $\}$
    (l) {Declaraciones de variables en C}
    (m) {Programas en C}

16. Pruebe que los lenguajes regulares son cerrados bajo: complemento, unión e intersección.

17. Probar que los lenguajes regulares son cerrados bajo concatenación y cerradura de Kleene.

18. Realizar las transformaciones indicadas usando los algoritmos vistos en clase.
    (a) Transformar los AFN's de la Figura 1 en AFD's
    (b) Transformar los AFD's de la Figura 2 en expresiones regulares.

(c) Transformar las siguientes expresiones regulares en AFN's

a' $(0+1)^*00(0+1)^*$

b' $(0+\varepsilon)(1+10)^*$

c' $(0+1)^*0(0+1)^7$

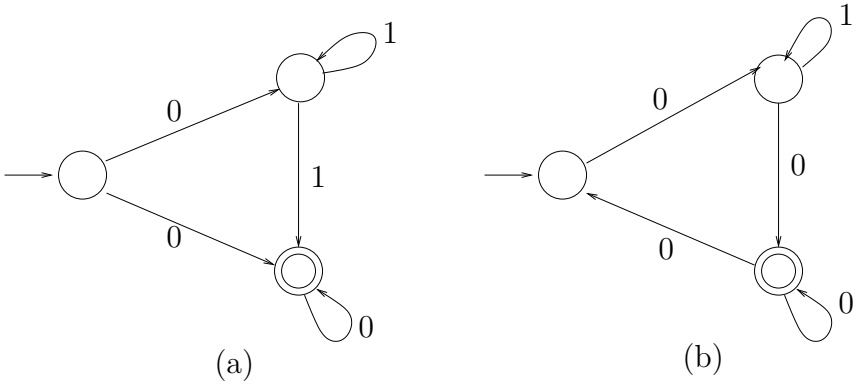

Figura 1: Convertir de AFN a AFD

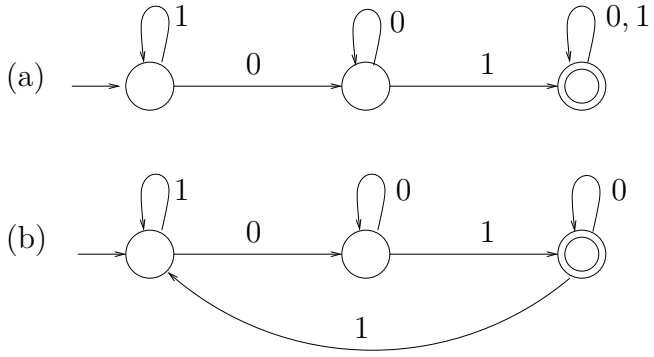

Figura 2: Convertir de AFD a expresión regular