Ma50b Complejidad Computacional	16 de Abril de 2009

Pauta Control 1

Prof. Cátedra: M. Kiwi — *Prof. Auxiliar: T. González, I. Fantini*

PROBLEMA 1:

(i).- Construiremos un autómata finito para decidir el lenguaje L de las secuencias $\omega \in \{0,1\}^*$ y luego usaremos la técnica vista en cátedra para asociar una expresión regular al autómata que sea equivalente al lenguaje que este reconoce. El autómata finito A que decide L se describe en la Figura 1.

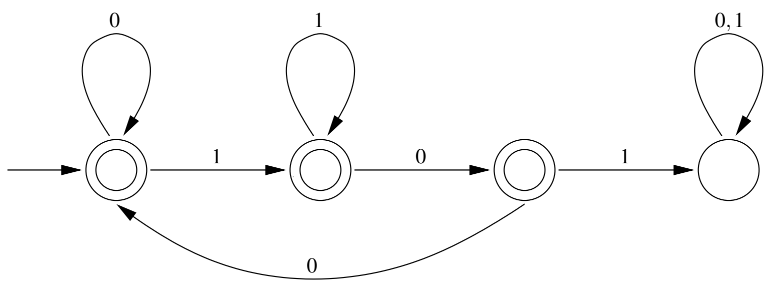

Figura 1: Autómata finito A que decide L.

En la Figura 2 se ilustra el proceso mediante el cual se deduce que L es generado por la expresión regular $(0+11^*00)^*(\varepsilon+11^*+11^*0)$.

(ii).- Afirmamos que L_1 no es regular. Para demostrarlo usaremos el Lema del Bombeo. Sea $p-1$ la constante de bombeo de L_1. Consideremos $s = 1^p01^p$. Claramente $s \in L_1$. Por Lema del Bombeo existen x, y, y z tales que $s = xyz$, $|y| > 0$, $|xy| \leq p$. Luego, x e y son necesariamente secuencias de puros 1's y en particular y no es una secuencia nula. Observar también que z contiene un único 0, luego $z = 1^a01^p$ donde $a = p-|x|-|y|$. Bombeando "hacia abajo" sigue que $xz \in L_1$. Pero $xz = 1^b01^p$ con $b = |x|+a = p-|y| < p$, pero la pertenencia de xz en L_1 implica que $p \leq b$, contradicción.

Afirmamos que L_2 es regular. En efecto, basta observar que $L_2 = \{1\omega : \omega \text{ contiene al menos un } 1\}$. Es fácil verificar que el autómata descrito en la Figura 3 reconoce L_2.

Finalmente, a pesar de que aparentemente reconocer L_3 requeriría una cantidad de memoria no acotada, una inspección más cuidadosa muestra que cualquiera sea el prefijo de una secuencia en $\{0,1\}^*$, es imposible que la diferencia entre la cantidad de veces que aparece 01 y 10 sea mayor estricto que 1 (en valor absoluto). Luego, un autómata que pueda mantener registro de la mentada referencia puede reconocer L_3. El autómata de la Figura 4 hace lo recién descrito y se verifica que reconoce L_3.

(iii).- Afirmamos que S es regular. En efecto, sea A un autómata finito que reconoce L. Como se vió en cátedra, sin pérdida de generalidad podemos asumir que A posee un único estado de aceptación a costa quizás de suponer que A es un automata finito con ε-transiciones.

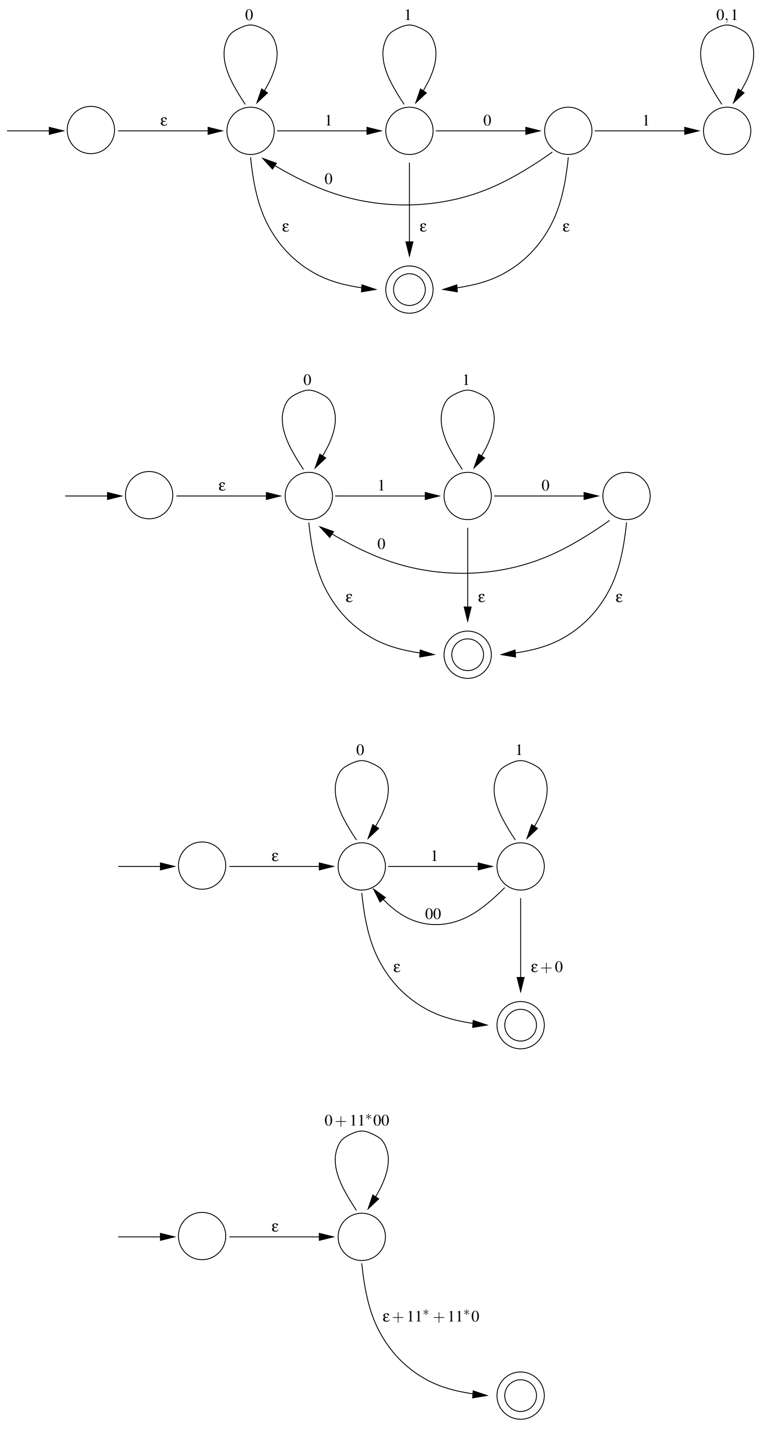

Figura 2: Proceso de generación de la expresión regular para L.

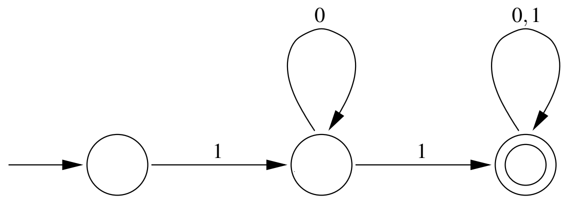

Figura 3: Autómata que reconoce L_2.

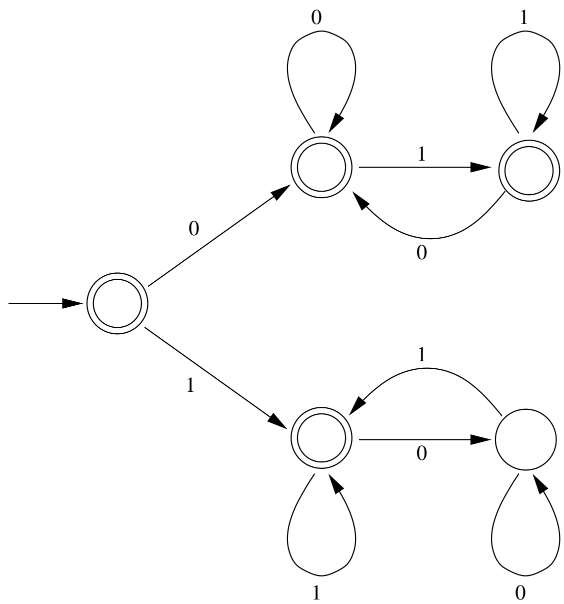

Figura 4: Autómata que reconoce L_3.

Sea $A^{\mathcal{R}}$ el automata finito con ε-transiciones que se obtiene de A dando vuelta los arcos en el diagrama de A. Se verifica (ejercicio) que $A^{\mathcal{R}}$ reconoce el conjunto de secuencias $\omega^{\mathcal{R}}$ con $\omega \in L$, que en lo sucesivo denotaremos $L^{\mathcal{R}}$.

Sean Q, s, y q_a el conjunto de estados, el estado de partida, y el único estado de aceptación de A, respectivamente. Consideramos entonces el autómata "producto" B con conjunto de estados $Q \times Q$ con los cuales B "simula" en paralelo los autómatas A y $A^{\mathcal{R}}$. Concretamente, cualquiera que sea $a \in \Sigma_\varepsilon$

$$\delta_B((q,q'),a) = \delta_A(q,a) \times \delta_{A^{\mathcal{R}}}(q,a)\,.$$

El estado de partida de B es (s,q_a) y sus estados de aceptación son los (q,q) tal que $q \in Q$.

Afirmamos que B reconoce S. En efecto, B acepta ω si y solo si existe un $q \in Q$ tal que $q \in \delta_A(s,\omega)$ y $q \in \delta_{A^{\mathcal{R}}}(q_a,\omega)$. Pero $q \in \delta_{A^{\mathcal{R}}}(q_a,\omega)$ si y solo si $q_a \in \delta_A(q,\omega^{\mathcal{R}})$. Luego, si B acepta ω, entonces $q_a \in \delta_A(s,\omega\omega^{\mathcal{R}})$, es decir $\omega\omega^{\mathcal{R}} \in L$. Por otro lado, si $\omega\omega^{\mathcal{R}} \in L$ entonces existe un $q \in Q$ tal que $q \in \delta_A(s,\omega)$ y $q_a \in \delta_A(q,\omega^{\mathcal{R}})$. Equivalentemente, existe un $q \in Q$ tal que $q \in \delta_A(s,\omega)$ y $q \in \delta_{A^{\mathcal{R}}}(q_a,\omega)$, es decir $(q,q) \in \delta_B(s,q_a)$, luego B acepta ω.

PROBLEMA 2:

(i).- Sea L un lenguaje reconocible (respectivamente decidible) por una máquina M con izquierda reseteo. Sea M' la máquina de Turing standard que simula en un paso cada transición de M en que M mueve su cabeza lectora a la derecha. En cualquier otro caso, M' simula una transición $\delta_M(q,a) = (r,b,RESET)$ escribiendo primero b en la celda bajo la cabeza lectora, desplazando después su cabeza lectora hasta la primera celda de su cinta, y finalmente entrando en el estado r. Es fácil ver que M y M' reconocen el mismo lenguaje. Sigue que L es reconocible (respectivamente decidible).

Supongamos ahora que L es es un lenguaje reconocible (respectivamente decidible) por una máquina de Turing standard M. Describiremos una máquina de Turing M' con izquierda reseteo que simula M. Informalmente, M' es un máquina que simula un movimiento a la izquierda de la cabeza lectora de M, vía ir desplazando una marca * hacia la derecha hasta detectar que la celda a la derecha de dicha marca también está marcada, en cuyo caso borra la segunda marca, hace un $RESET$, y desplaza su cabeza lectora hasta encontrar la marca *. Formalmente,

M' = En la entrada ω

Mientras la simulación de M en ω no se detenga:

$(r,b,D) \leftarrow \delta_M(q,a)$ donde q y a son el estado en que se encuentraría M y el caracter bajo su cabeza lectora

Si $D = R$

Escribir b, recordarse el estado r en que quedaría M, y mover la cabeza una celda a la derecha

En caso contrario,

Escribir $b^\star$, recordarse el estado r en que qeudaría M y hacer un $RESET$

(#) Reemplazar el caracter, digamos a, bajo la cabeza lectora por $a^\star$ y moverse una celda a la derecha

Si la celda bajo la cabeza lectora no contiene un *,

Hacer un $RESET$, desplazarse a la derecha hasta encontar una celda que contiene *, digamos $c^\star$, reemplazar su contenido por c y mover la cabeza lectora una celda a la derecha. Ir a (#).

En caso contrario,

Reemplazar el caracter bajo la cabeza lectora, específicamente $b^\star$, por b, hacer un $RESET$, y desplazar la cabeza lectora a la derecha hasta encontrar la celda que contiene *

Aceptar/Rechazar si r es un estado de aceptación/rechazo de M

Se verifica (ejercicio) que M' y M reconocen el mismo lenguaje. Sigue que todo lenguaje reconocible (respectivamente decidible) es reconocible (respectivamente decidible) por una máquina de Turing con izquierda reseteo.

(ii).- Veamos primero que L es co-reconocible. En efecto, consideremos la siguiente máquina de Turing

N = En la entrada $\langle M,M'\rangle$, M y M' máquinas de Turing

Para $t = 0,1,2,\ldots$

Para cada ω entre las lexicográficamente primeras palabras en Σ_M^*

Simular en paralelo t pasos de M y M' en ω y aceptar si ambas máquinas aceptan

Se verifica que N reconoce L (ejercicio).

Recordemos que $Halt_{mT} = \{\langle M, \omega\rangle : M$ se detiene en $\omega\}$. En cátedra se vió que $Halt_{mT}$ es indecidible. Como $Halt_{mT}$ es claramente reconocible, por resultado visto, sigue que $\overline{Halt_{mT}}$ no es reconocible. Veremos ahora que $\overline{Halt_{mT}}$ mucho a uno reduce a L, de donde se concluye que L no es reconocible. Sea T la mq́uina de Turing que acepta cualquiera sea su entrada. Sea $S_{M,\omega}$ la máquina de Turing que en la entrada x, simula M en ω, y si M se detiene, entonces acepta x. Claramente $L(S_{M,\omega}) = \Sigma^*$ si M se detiene en ω y $L(S_{M,\omega}) = \emptyset$ en caso contrario. Consideremos la función f que a $\langle M, \omega\rangle$ con M máquina de Turing le asocia $\langle S_{M,\omega}, T\rangle$, y donde $f(\sigma) = \#$ si σ no es de la forma $\langle M, \omega\rangle$ con M máquina de Turing. Claramente, f es totalmente recursiva, y además

$$\langle M, \omega\rangle \in \overline{Halt_{mT}} \iff L(S_{M,\omega}) = \emptyset \iff L(S_{M,\omega}) \cap L(T) = \emptyset \iff \langle S_{M,\omega}, T\rangle \in L.$$

En resumen, $\overline{Halt_{mT}} \leq_m L$.

(iii).- Sea E un enumerador para A. Sea R la máquina de Turing tal que

R = En la entrada ω

Simula E hasta que este genera la ω-ésima salida $\langle D_\omega\rangle$ (ω visto como un número en binario).

Simula D_ω en ω

– Acepta, si D_ω rechaza ω

– Rechaza, si D_ω acepta ω

Cualquiera sea la entrada ω, dado que A es un lenguaje infinito, se generará una ω-ésima descripción $\langle D_\omega\rangle \in A$. Notar que D_ω se detiene cualquiera sea su entrada. Luego, la simulación de D_ω en ω se detiene. Sigue que R es una máquina que se detiene cualquiera sea la entrada y por lo tanto reconoce un lenguaje $L(R)$ que es decidible.

Afirmamos que $\langle R\rangle \notin A$. En efecto, si $\langle R\rangle \in A$, entonces $R = D_{\omega_0}$ para algún ω_0. Sigue que en la entrada ω_0, la máquina $R = D_{\omega_0}$ acepta si $D_{\omega_0} = R$ rechaza ω_0, y que $R = D_{\omega_0}$ rechaza si $D_{\omega_0} = R$ acepta ω_0.