Introducción a la Informática Teórica
Tarea #1
"La Maldición del Tercer Ojo de Visnú"

Hernán Vargas Leighton, 201073009-3

2 de abril 2014

Respuestas

1. Para satisfacer las tres condiciones, necesitamos un autómata capaz de detectar las secuencias 44 y 23, además, debe ser capaz de diferenciar cuando el *string* tiene un número par de $5's$. Para ello digamos:

   $M = (Q, \Sigma, \delta, q_0, F)$

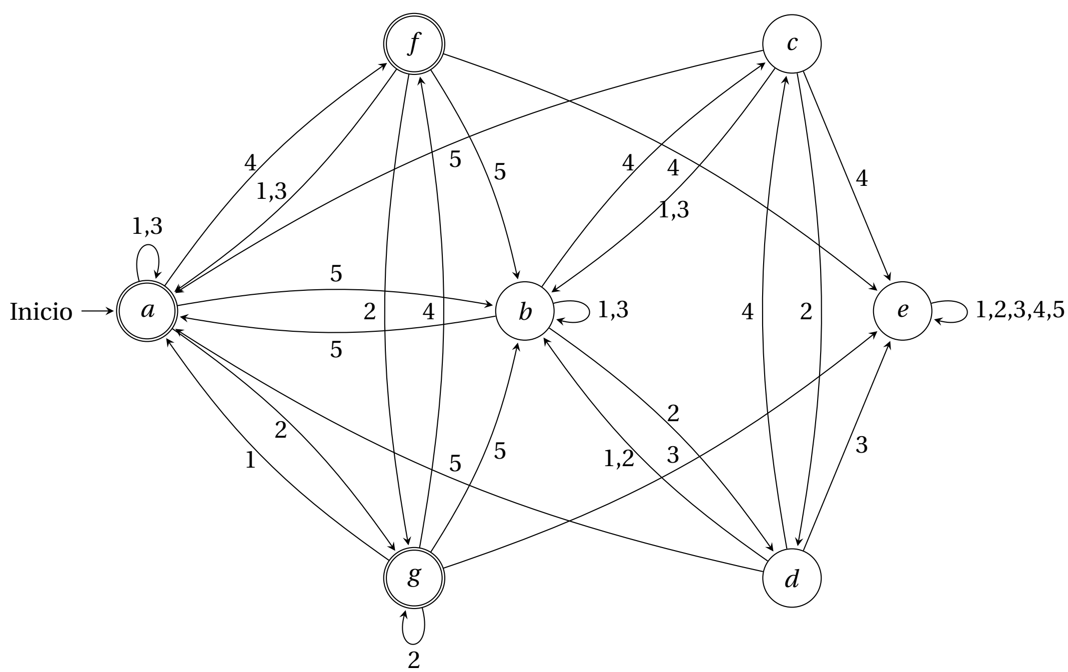
   Con:

   $Q = \{a, b, c, d, e, f, g\}$ $\Sigma = \{1, 2, 3, 4, 5\}$ $q_0 = a$ $F = \{a, f, g\}$

   Y con δ: función de transición. Con ello tenemos que la tabla de transiciones será:

estado / entrada	a	b	c	d	e	f	g
1	a	b	b	b	e	a	a
2	g	d	d	b	e	g	g
3	a	b	b	e	e	a	e
4	f	c	e	c	e	e	f
5	b	a	a	a	e	b	b

   Notar que si se llega al estado e ya no podremos salvarnos de la maldición de Visnú.
   Con la tabla hecha nos basta escribir el autómata determinista:



Sobre los estados finales podemos decir los siguiente:

- Estado a**:** Para *strings* terminados en 1, 3 o 5 (par).
- Estado f**:** Para *strings* terminados en 4.
- Estado g**:** Para *strings* terminados en 2.

Cabe destacar que los estados c y d son variantes para número de 5 impar de los estados f y g respectivamente, en este caso no se cumplen las condiciones.
Además el estado e es al cual se cae cuando ya no es posible evitar la maldición.

2. En primer lugar definiré nuestro alfabeto con las iniciales de Ammavaru, Brahma y Chandra:

$$\Sigma = \{A, B, C\}$$

Se debe cumplir con:

a) El *string* debe tener un largo par.
b) El *string* debe contener al menos un AB.
c) El *string* no debe tener CC.

Para cumplir con estas reglas debemos:

a) Construir la expresión en con base en palabras de largo 2.
b) Escribir la expresión regular AB. Como su largo es 2 no interfiere con la regla a).
c) Asegurar que ninguna palabra de largo 2 sea CC. Además se debe tener la precaución que una palabra terminada en C no esté junto a una palabra que comience con C y viceversa.

Podemos construir una expresion regular que cumpla con estas tres reglas definiendo las siguientes palabras:

$$a = (A|B)(A|B) \qquad b = CA \qquad c = CB \qquad d = AC \qquad e = BC \qquad f = CC$$

Inmediatamente notamos que a serán las palabras sin problemas *incluida* AB. Por otro lado b, c, d, e son las que debemos tener en cuenta y f no debe estár incluida.
Entonces:

- $(a|b|c)$ denota una palabra que **no termina en** C.
- $(d|e)^+a$ denota una palabra que **no comienza con** C**, no termina en** C **y no tiene** CC **intermedias pero si puede tener** C.
- $((a|b|c)|(d|e)^+a)^*$ está compuesta por las anteriores y denota toda palabra de largo par sin CC que **no termine en C**
- Para aceptar la opción de que el *string* termine en C (sin quebrar las otras reglas) nos basta con agregar $(d|e)$ o (ϵ) al final de la expresión actual.
  Convenientemente escribimos: $(((a|b|c)|(d|e)^+a)^*)(\epsilon|(d|e))$
- Por último, para asegurar que exista al menos un AB agregamos directamente un AB y lo *rodeamos* por la expresion anterior. Como resultante tenemos: $(((a|b|c)|(d|e)^+a)^*)(\epsilon|(d|e))AB(((a|b|c)|(d|e)^+a)^*)(\epsilon|(d|e))$

Ahora nos basta volver al alfabeto inicial. Para ello tenemos que:

$$(a|b|c) = (A|B|C)(A|B) \qquad\qquad (d|e) = (A|B)C$$

Además de las definiciones. Con esto tenemos que la expresión regular que cumple con las tres reglas y por lo tanto denota los poderosos talismanes de Visnú es:

$$\left(\left((A|B|C)(A|B)\right)|\left(((A|B)\,C)^+(A|B)(A|B)\right)\right)^*\left(\epsilon|(A|B)C\right)AB\left(\left((A|B|C)(A|B)\right)|\left(((A|B)\,C)^+(A|B)(A|B)\right)\right)^*\left(\epsilon|(A|B)C\right)$$

3. Las tres condiciones son:

   a) Numero impar de $1's$.
   b) Si hay $0's$ serán $(011)'s$.
   c) El *string* siempre termina en 1

   Gracias a las reglas *a*) y *c*) podemos simplemente buscar un *string* de largo par y agregar un 1 al final.
   Este nuevo *string* estara denotado por:

   - $(011)^*$ si está compuesto solo de $(011)'s$.
   - $(1(011)^*1)^*$ si son solo $(11)'s$ o si $(011)'s$ estan rodeados por un 1.

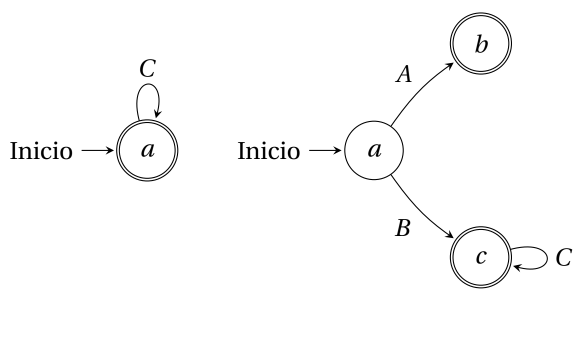
   Notar que éstas reglas cumplen con mantener la cantidad par de $1's$, además:

   $$\left(011|1\,(011)^*\,1\right)^*$$

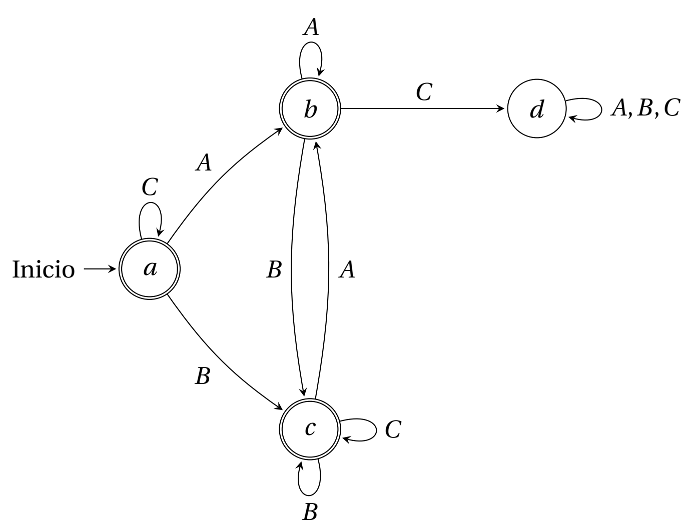
   Aceptará cualquier conbinación de $(011)'s$ y $1's$ con un número par de $1's$. Nos basta solo agregar un 1 al final de está expresión regular, pero éste 1 puede ser en la forma de (1) o (011) es decir: (1|011).
   Con esto tenemos que la expresión regular que representa la información de los rayos del tercer ojo de Visnú es:

   $$\left(011|1\,(011)^*\,1\right)^*\left(1|011\right)$$

4. Digamos: $\Sigma = \{A, B, C\}$, buscamos un autómata finito determinista $M = (Q, \Sigma, \delta, q_0, F)$ que cumpla con: $\mathscr{L}(R) = C^*(A|BC^*)^*$.
   Sabemos que C^*, $A|BC^*$ y $(A|BC^*)^*$ son respectivamente:



   Con esto nos basta hacer la unión y agregar las *entradas* faltantes para obtener el autómata finito determinista resultante:



Sobre los estados finales podemos decir los siguiente:

- Estado a**:** Para *strings* compuestos exclusivamente de $C's$.
- Estado b**:** Para *strings* terminados en A.
- Estado c**:** Para *strings* terminados en B o C.

Cabe destacar que el estado d es alcanzado cuando la cadena no es sagrada, es decir cuando el *string* no calza con la expresión regular $\mathscr{L}$
Ahora podemos definir $M = (Q, \Sigma, \delta, q_0, F)$ con:

$$Q = \{a, b, c, d\} \qquad \Sigma = \{A, B, C\} \qquad q_0 = a \qquad F = \{a, b, c\}$$

Y con δ: función de transición. Con ello tenemos que la tabla de transiciones será:

entrada \ estado	a	b	c	d
A	b	b	b	d
B	c	c	c	d
C	a	d	c	d

HV/$\LaTeX 2_\varepsilon$