



Tecnológico de Costa Rica

Centro Académico San José

Escuela de Ingeniería en Computación

Compiladores e Intérpretes

I Semestre 2016

Apuntes Clase - 17 de Marzo del 2017

Profesor: Dr. Francisco J. Torres-Rojas

Apuntadora: Liza Chaves Carranza, 2013016573

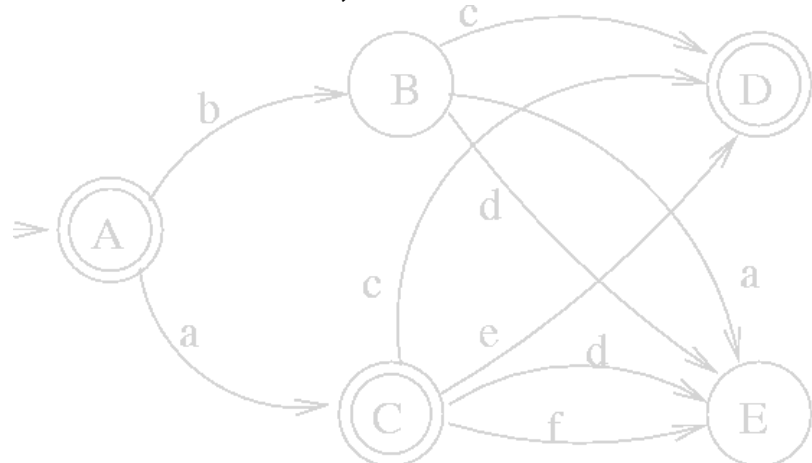


Tabla de Contenidos

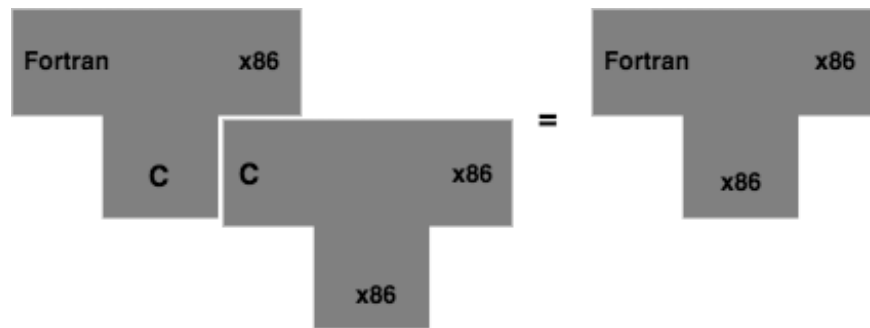
Recordatorios	3
Repaso Sobre Diagramas T	3
Ejemplos de Autómatas Determinísticos de Estados Finitos – Cont.	4
Ejemplo 11.....	4
Ejemplo 12.....	4
Ejemplo 13.....	5
Ejemplo 14.....	5
Características	6
Definición Formal.....	6
Ejemplo 1	7
Ejemplo 2	7
Ejemplo 3	8
Implementación de DFAs.....	8
Ejemplo de Código.....	9
Ejemplo de Uso	9
DFA y Lenguajes	11
Nuestra Misión	11
Diseño de DFAs.....	12
Conceptos Básicos	13
Ejemplos	13
Ejemplo 1	13
Ejemplo 2	14
Ejemplo 3	14
Ejemplo 4	15
Ejemplo 5	16
Ejemplo 6	16
Ejemplo 7	17
Ejemplo 8	18
Ejemplo 9	18
Ejemplo 10.....	19

Recordatorios

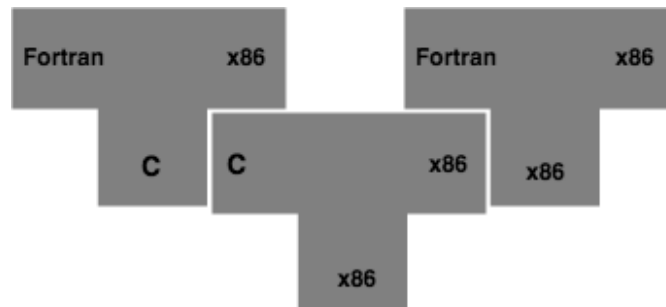
- Es muy probable que el examen sea el viernes antes de salir a semana santa (7 de Abril), dependiendo de qué tan cómodo se sienta el profesor con la materia que hemos visto.

Repaso Sobre Diagramas T

En la clase donde revisamos los diagramas T, utilizamos la notación:



Pero la notación que utilizan algunos libros (y que podrá salir en preguntas del examen) es la siguiente:

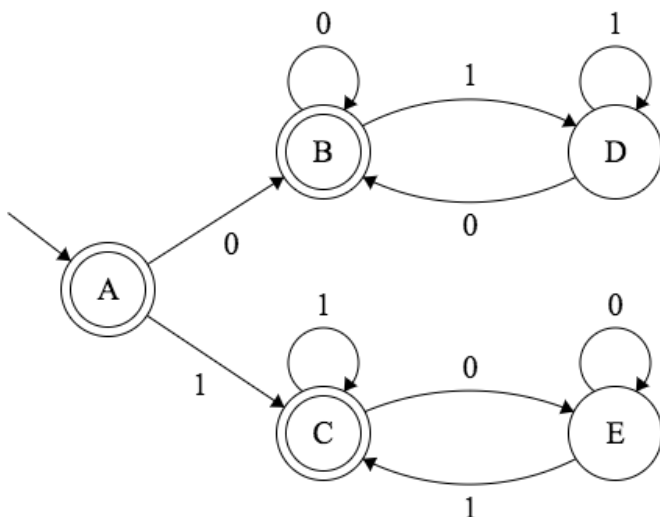


Donde el resultado sería el último diagrama T de la derecha. Los compiladores pueden “encaramarse” uno en otro para formar nuevos compiladores, y utilizar los que ya se han creado previamente. Siempre se debe de buscar la forma siguiente para saber cuáles son los compiladores involucrados:



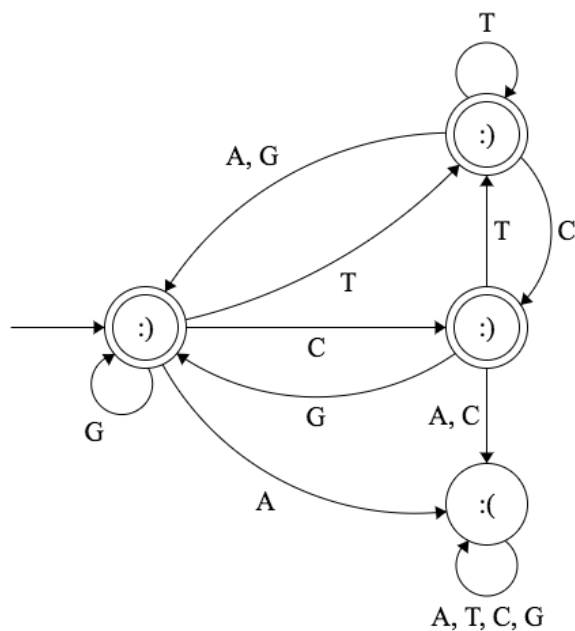
Ejemplos de Autómatas Determinísticos de Estados Finitos – Cont.

Ejemplo 11

¿Acepta o rechaza w ? $w = 000101011101$ 

¡Se rechaza la hilera!

Ejemplo 12

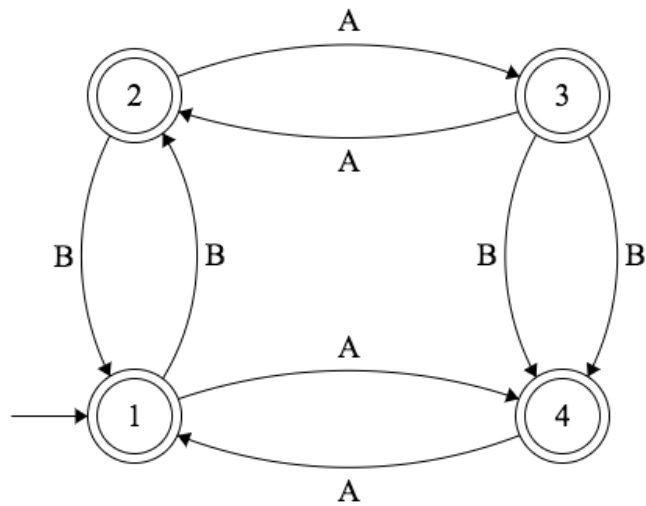
¿Acepta o rechaza w ? $w = TATAGTACGTTTCG$ 

¡Se acepta la hilera!

Ejemplo 13

¿Acepta o rechaza w ?

$w = \varepsilon$

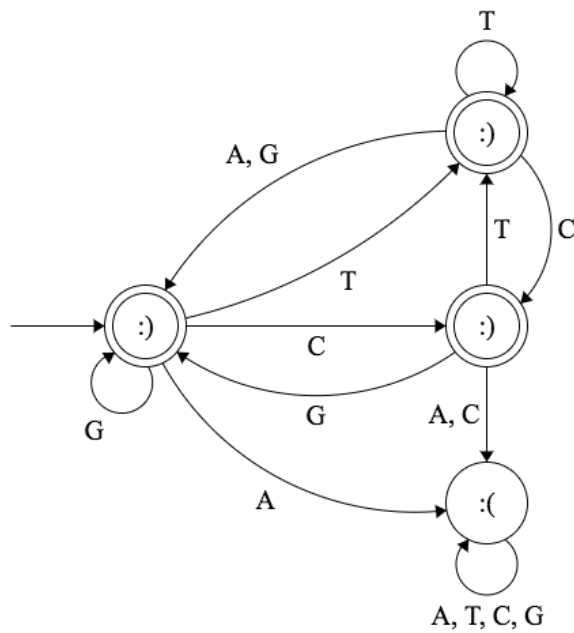


¡Se rechaza la hilera!

Ejemplo 14

¿Acepta o rechaza w ?

$w = \varepsilon$



¡Se acepta la hilera!

Características

- Autómata Determinístico de Estados Finitos – **Máquina**.
- En inglés: *Deterministic Finite Automaton*.
- De ahora en adelante los llamaremos **DFA**.
- ¿Finitos? Si, hay un número **fijo y finito** de estados posibles.
- ¿Determinístico?
 - Dado un **estado** y un **símbolo**, hay un **único** estado siguiente, se sabe de manera precisa a dónde se debe ir.
 - Dada una entrada siempre vamos a poder predecir a dónde vamos a terminar.



Notas:

- La hilera que el autómata recibe tiene que recorrerse completamente para ser aceptada, pero esto no es necesario para que sea rechazada.
- Si algún estado no tiene más caminos, se rechaza la hilera inmediatamente.



¡Agárrense del Sombrero!

Definición Formal

Un Autómata Determinístico de Estados Finitos (DFA) M es un quinteto $M = (Q, \Sigma, \delta, q_0, F)$, donde:

- Q es un conjunto finito de **estados**.
- Σ es un **alfabeto**.
- $\delta: Q \times \Sigma \rightarrow Q$ es la **función de transición** (\times equivale a producto cartesiano).
- $q_0 \in Q$ es el **estado inicial**.
- $F \subseteq Q$ conjunto de **estados de aceptación** (este conjunto podría estar vacío).

Ejemplo

1

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

- $Q = \{a, b, c, d\}$

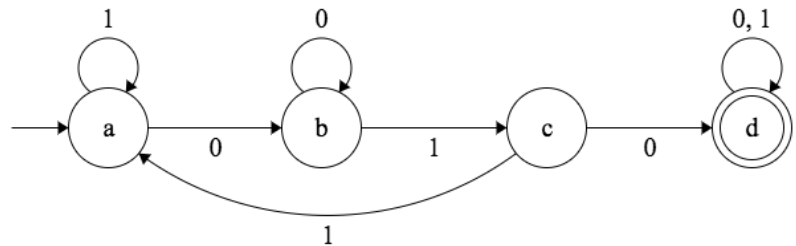
- $\Sigma = \{0, 1\}$

- $\delta =$

	0	1
a	b	a
b	b	c
c	d	a
d	d	d

- $q_0 = a$

- $F = \{d\}$



Ejemplo 2

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

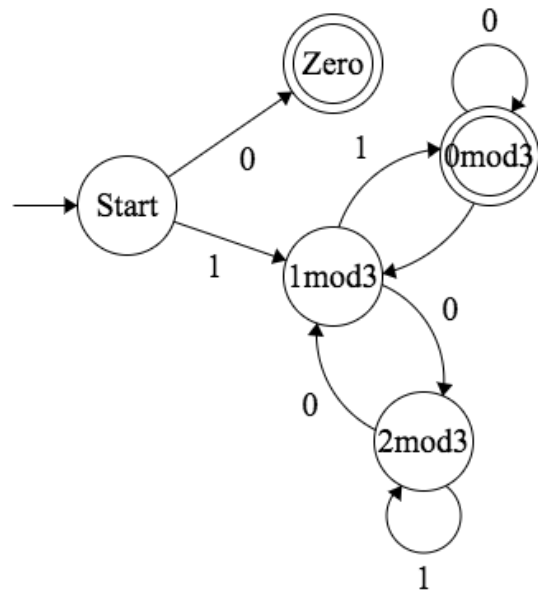
- $Q = \{\text{start}, \text{zero}, 0\text{mod}3, 1\text{mod}3, 2\text{mod}3\}$

- $\Sigma = \{0, 1\}$

- Estado inicial (q_0): start

- Estados de finalización (F): zero, 1mod3

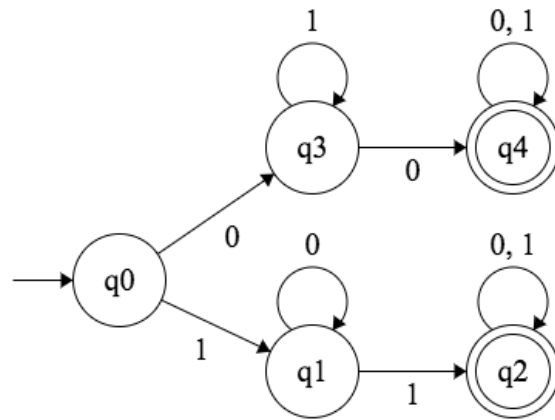
- Transiciones: ninguna.



Ejemplo 3

- $M = (Q, \Sigma, \delta, q_0, F)$:
- $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- $\Sigma = \{0, 1\}$
- $\delta =$

	0	1
q0	q3	q1
q1	q1	q2
q2	q2	q2
q3	q4	q3
q4	q4	q4



- $q_0 = q_0$
- $F = \{q_2, q_4\}$

Nota:

- Cuando todos los símbolos del alfabeto son aceptados para pasar de un estado a otro, no se ponen todos los símbolos, se escribe Σ para mayor comodidad.

Implementación de DFAs

¿Cómo lo programamos?

El algoritmo siempre es “casi” el mismo, lo que cambia son las transiciones (tabla), las partes del DFA acoplado a la tabla son:

- **Filas:** estados.
- **Columnas:** símbolo.
- **Contenido:** siguiente estado.

3	1	0
1	2	0
2	2	1
4	3	0
4	4	1

Puede existir un vector de aceptación/rechazo y también se puede tener un *driver* general para DFAs.

Argumentos que recibirá la función del driver del DFA:

- Tabla de Transiciones.
- Vector de aceptación/rechazo.
- Puntero a función de codificación de símbolo.
- Hilera a ser revisada.
- Estado inicial del autómat.

Esta función regresa el resultado como “aceptación” o “rechazo” (1 o 0).



Mameluco que forchan conseguirá próximamente.



El hijo perdido de forchan

Ejemplo de Código

```
int DFA_driver (int **Table,
                int * Accept,
                int (* code) (char c),
                char * string,
                int state)
{
    char * s;
    int k;

    s = string;
    k = state;
    while (* s)
        k = Table[k][code(*s++)];

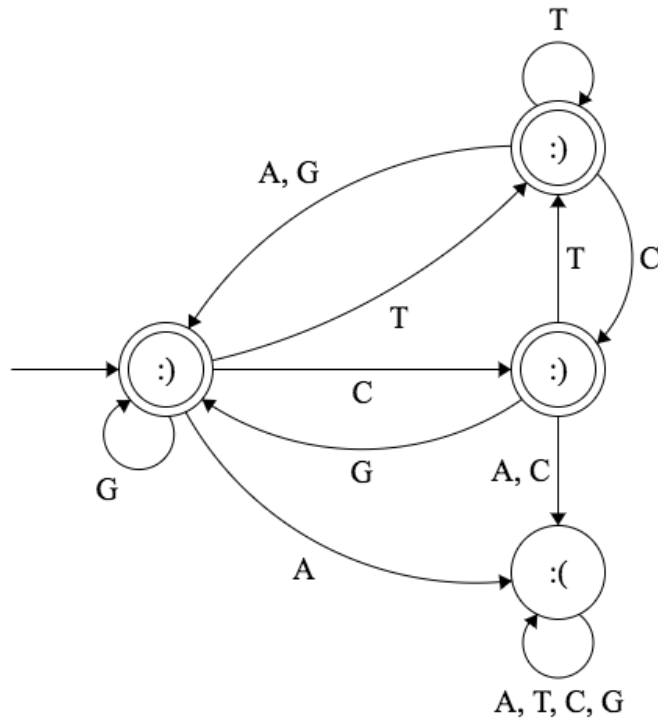
    return(Accept[k]);
}
```



Aclaración: *s se utiliza para recorrer la hilera mientras no esté en null.

Ejemplo de Uso

Para este DFA tomaremos la información y la introduciremos en el código escrito previamente.



	A	T	C	G
0	3	1	2	0
1	0	1	2	0
2	3	1	3	0
3	3	3	3	3

0	1
1	0
2	0
3	0



Se pasan los datos a estructuras de código, y finalmente se le pasan como argumentos a la función DFA_driver.

```
int tabla[4][4] = {{3, 1, 2, 0},{0, 1, 2, 0},
                  {3, 1, 3, 0},{3, 3, 3, 3}};
int accept[4] = {1, 1, 1, 1};

int code(char c)
{
    if (c == 'A') return 0;
    if (c == 'T') return 1;
    if (c == 'C') return 2;
    if (c == 'G') return 3;
}

...
DFA_driver (tabla, accept, code, hilera, 0);
...
```

DFA y Lenguajes

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

Sea $w = w_1, w_2, \dots, w_n$ una hilera sobre Σ

M acepta w si existe una secuencia de estados $r_0, r_1, r_2, \dots, r_n$ tal que:

- Todos los r_i pertenecen a Q .
- $r_0 = q_0$
- $\delta(r_i, w_{i+1}) = r_{i+1}$ para $i = 0, \dots, n-1$ (matemáticamente hermoso).
- $r_n \in F$

La máquina M reconoce al lenguaje L si y solo si:
 $L = \{w \mid M \text{ acepta } w\}$

Nuestra Misión

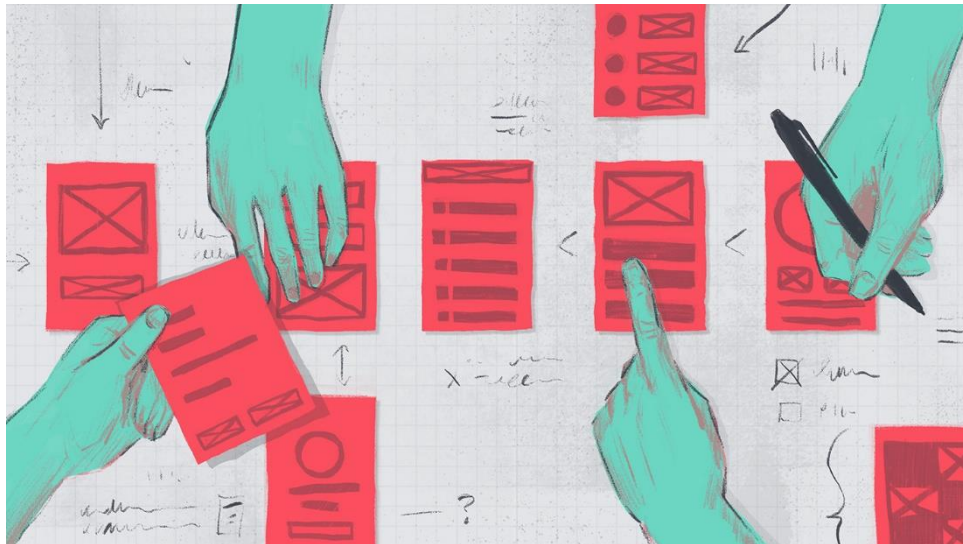


- Diseñar DFAs que reconozcan lenguajes.
- Aunque no nos demos cuenta, los DFAs están en todos los scanners de un compilador típico.
- Se puede automatizar (con flex y bison) pero estamos sometidos a la regla draconiana de que no se puede usar nada si no se sabe de dónde proviene.



¡Agárrense del sombrero! (de nuevo)

Diseño de DFAs



Misión: nos describen un lenguaje L y hacemos un DFA que lo reconozca.

El DFA tiene que aceptar **todas** las hileras de L . (Si falta alguna hilera de L , el autómata está malo).

El DFA **no puede** aceptar hileras que **no sean miembros** de L . (Si acepto alguna hilera que en realidad no es parte L , el autómata está malo).

Conceptos Básicos

- Número finito de estados. (Los estados son nuestros amigos) (también si llevan 47814 estados en el examen y no terminan, algo están haciendo mal).
- Las etiquetas de los estados, también son nuestras amigas, no son obligatorias pero se ponen para saber que es lo que sé sobre el estado o arco (por ejemplo “he visto dos ceros”).
- En principio, de cada estado salen las transiciones con todos los elementos de Σ . Muchas veces no se ponen flechas porque sabemos que ya se va a rechazar una hilera, entonces se ahorra esa parte.
- ¿Cuáles estados (situaciones) se deben aceptar?
- ¿Es ϵ parte de L ? (esto nos dice si el estado inicial corresponde a uno de aceptación o de rechazo, normalmente es lo primero que hay que cuestionarse cuando comenzamos a diseñar un DFA).

Ejemplos

Ejemplo 1

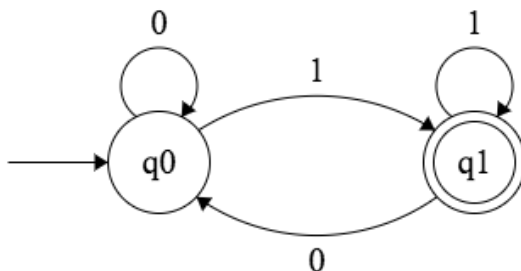
Sea L el lenguaje sobre $\Sigma = \{0, 1\}$ de hileras que terminan en 1.

Diseñe un DFA que reconozca L .

Primer paso:

¿Es ϵ parte de $\Sigma = \{0, 1\}$?

R/ No, porque no termina en 1, por lo tanto el estado inicial es de rechazo.



Funcionamiento:

1. Si lee un 0, se rechaza y se mantiene en q_0 , si lee un 1 pasa a un estado de aceptación y se mueve a q_1 .
2. Si vuelve a leer un 1, se mantiene en estado de aceptación y se mantiene en q_1 , de lo contrario se devuelve a q_0 y se pasa a un estado de rechazo.

Ejemplo 2

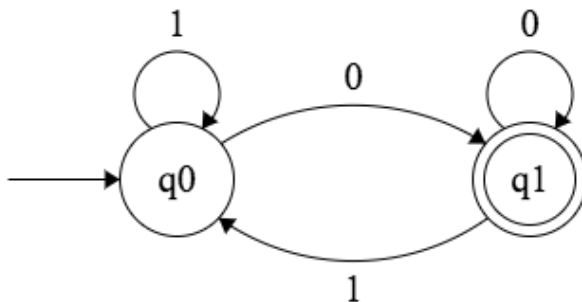
Sea **L** el lenguaje sobre $\Sigma = \{0, 1\}$ de hileras que terminan en 0.

Diseñe un DFA que reconozca **L**.

Primer paso:

¿Es ϵ parte de $\Sigma = \{0, 1\}$?

R/ No, porque no termina en 0, por lo tanto el estado inicial es de rechazo.

**Funcionamiento:**

- Si lee un 1, se rechaza y se mantiene en q_0 , si lee un 0 pasa a un estado de aceptación y se mueve a q_1 .
- Si vuelve a leer un 0, se mantiene en estado de aceptación y se mantiene en q_1 , de lo contrario se devuelve a q_0 y se pasa a un estado de rechazo.

Ejemplo 3

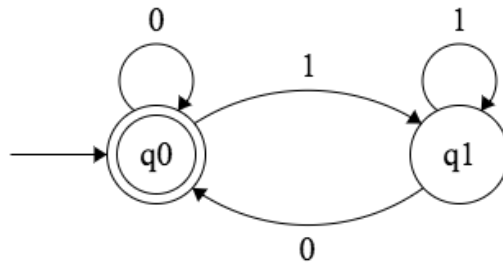
Sea **L** el lenguaje sobre $\Sigma = \{0, 1\}$ de hileras que **no terminan** en 1.

Diseñe un DFA que reconozca **L**.

Primer paso:

¿Es ϵ parte de $\Sigma = \{0, 1\}$?

R/ Si, porque ϵ es una hilera sin símbolos, por lo tanto el estado inicial es de aceptación.

**Funcionamiento:**

1. Si lee un 0, entra en estado de aceptación y se mantiene en q0, si lee un 1 se mueve a q1 y entra en estado de rechazo.
2. Si vuelve a leer un 0 se devuelve a q0, y pasa al estado de aceptación.

Ejemplo 4

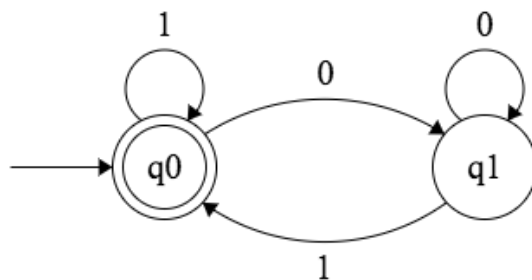
Sea **L** el lenguaje sobre $\Sigma = \{0, 1\}$ de hileras que **no terminan** en 0.

Diseñe un DFA que reconozca **L**.

Primer paso:

¿Es ϵ parte de $\Sigma = \{0, 1\}$?

R/ Si, porque ϵ es una hilera sin símbolos, por lo tanto el estado inicial es de aceptación.

**Funcionamiento:**

1. Si lee un 1, entra en estado de aceptación y se mantiene en q0, si lee un 0 se mueve a q1 y entra en estado de rechazo.
2. Si vuelve a leer un 1 se devuelve a q0, y pasa al estado de aceptación.

Ejemplo 5

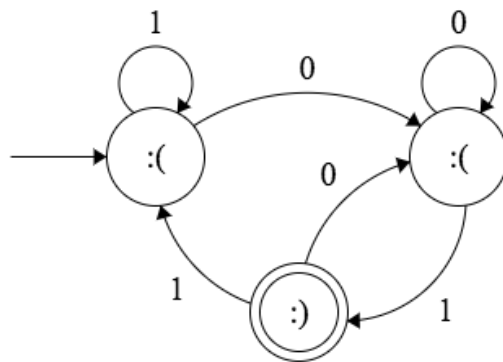
Sea **L** el lenguaje sobre $\Sigma = \{0, 1\}$ de hileras que **terminan** en 01.

Diseñe un DFA que reconozca **L**.

Primer paso:

¿Es ε parte de $\Sigma = \{0, 1\}$?

R/ No, porque no termina en 01, por lo tanto el estado inicial es de rechazo.



Funcionamiento:

1. Si lee un 1, entra en estado de rechazo, si leyera un 0 se mueve al estado siguiente y continúa en estado de rechazo, si continúa leyendo ceros, seguirá en rechazo.
2. Si vuelve a leer un 1, se mueve al estado siguiente (:)) y cambia a estado de aceptación, si luego de esto leyera un 0, se devuelve al estado de “he leído un 0” y pasa a rechazo, y así sucesivamente hasta aceptar o rechazar la hilera.

Ejemplo 6

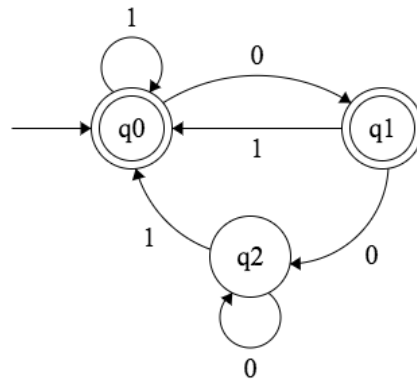
Sea **L** el lenguaje sobre $\Sigma = \{0, 1\}$ de hileras que **no terminan** en 00.

Diseñe un DFA que reconozca **L**.

Primer paso:

¿Es ε parte de $\Sigma = \{0, 1\}$?

R/ No, porque no termina en 00, por lo tanto el estado inicial es de rechazo.

**Funcionamiento:**

1. Si lee un 1, entra en estado de aceptado q0, si leyera un 0 se mueve al estado q1 de aceptación también.
2. Si continúa leyendo y encontrase un 0, pasa al estado q2 de rechazo, y si sigue leyendo ceros, se queda en ese estado.
3. Si encontrara otra vez un 1, pasa al estado inicial q0 y se acepta la hilera de nuevo.

Ejemplo 7

Sea **L** el lenguaje sobre $\Sigma = \{A, T, C, G\}$ de hileras cuya longitud sea múltiplo de 3.

Diseñe un DFA que reconozca **L**.

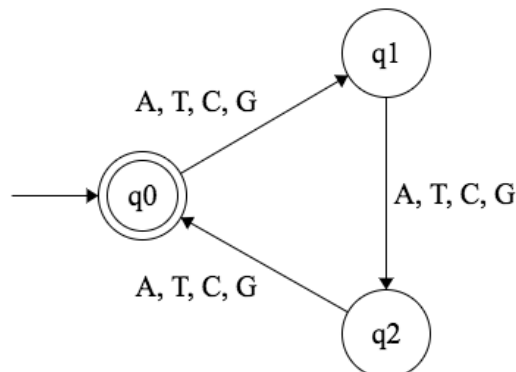
Primer paso:

¿Es ϵ parte de $\Sigma = \{A, T, C, G\}$?

R/ Si, porque la longitud de ϵ es 0 y este es múltiplo de 3, entonces el estado inicial es de aceptación.

Nota:

- Se puede utilizar el símbolo Σ para las transiciones, equivale a cualquier miembro del lenguaje Σ que se lea.



Funcionamiento:

1. Si lee cualquier hilera, el estado q_1 es de aceptación, puesto que lo que importa es el largo de la hilera que se haya leído, y como son 3 estados, siempre la longitud de la hilera que se lea en el estado q_0 va a ser múltiplo de 3, mientras que los demás, no.

Ejemplo 8

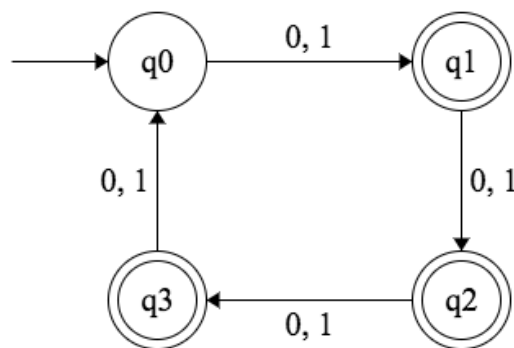
Sea L el lenguaje sobre $\Sigma = \{0, 1\}$ de hileras cuya longitud **no sea** múltiplo de 4.

Diseñe un DFA que reconozca L .

Primer paso:

¿Es ε parte de $\Sigma = \{0, 1\}$?

R/ No, porque la longitud de ε es 0 y este es múltiplo de 4, entonces el estado inicial es de rechazo.

**Funcionamiento:**

1. Si lee cualquier hilera, el estado q_0 es de rechazo, puesto que lo que importa es el largo de la hilera que se haya leído, y como son 4 estados, siempre la longitud de la hilera que se lea en el estado q_0 va a ser múltiplo de 4, mientras que los demás, no.

Ejemplo 9

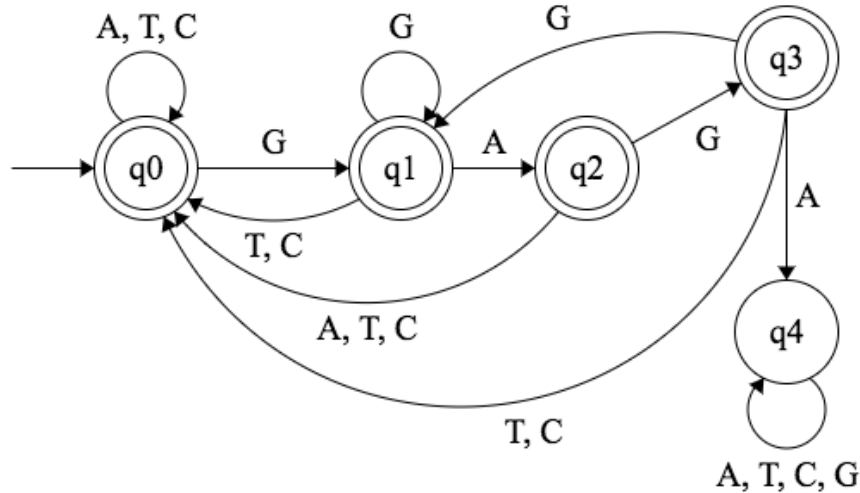
Sea L el lenguaje sobre $\Sigma = \{A, T, C, G\}$ de hileras que **no contengan** la subhilera “GAGA”.

Diseñe un DFA que reconozca L .

Primer paso:

¿Es ε parte de $\Sigma = \{A, T, C, G\}$?

R/ Si, porque ε es vacía y no contiene la subhilera “GAGA”, entonces el estado inicial es de aceptación.

**Funcionamiento:**

1. q4 es un estado que se denomina *sink*, porque todo lo que lea se pierde (no vale) y se rechaza la hilera.
2. Se lee una A, T o C se mantiene en el estado de aceptación q0, mientras lea una G se mueve al estado q1 ("he leído una G").
3. Si en el estado q1 vuelve a leer una A, pasa al estado de aceptación q2 y la hilera es aceptada, si lee T,C se devuelve al estado inicial q0.
4. Si sigue leyendo y encuentra una G, pasa al estado de aceptación q3, pero si lee A, T o C, se devuelve al estado q0, debido a que después de una A en este momento de la ejecución, tendría que seguir una G.
5. Si en el estado q3 se lee una G, se devuelve al estado q1, que es el que lee las G si se encuentra alguna que pueda ser parte de "GAGA". Pero si lee una A, la hilera se rechaza, ya que contiene la subhilera "GAGA" efectivamente.
6. Si en el estado q3 se lee T o C, se devuelve al estado q0, debido a que se descarta que sea la subhilera "GAGA" y tiene que seguir revisando.

Ejemplo 10

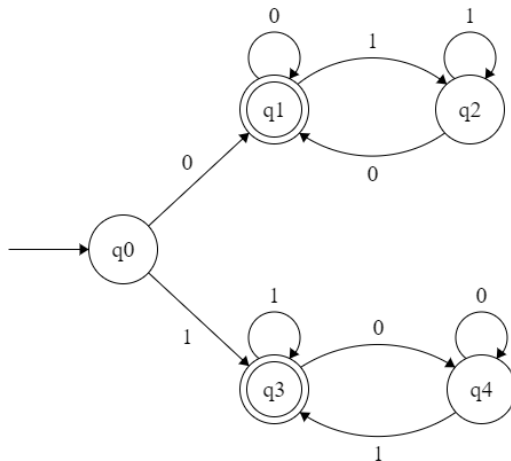
Sea **L** el lenguaje sobre $\Sigma = \{0, 1\}$ de hileras que **terminen** con el mismo símbolo con el que empezaron.

Diseñe un DFA que reconozca **L**.

Primer paso:

¿Es ϵ parte de $\Sigma = \{0, 1\}$?

R/ No, porque ϵ es vacía, no comienza ni termina con ningún símbolo, por lo tanto el estado inicial es de rechazo.

**Funcionamiento:**

1. Si lee un 1 o 0, entra en estado de rechazo q0.
2. Si continúa leyendo y encontrase un 0, pasa al estado q1 de aceptación, y si sigue leyendo 0, se queda en ese estado, si leyese un 1, pasa al estado de rechazo y así sucesivamente.
3. Si del estado q0 lee un 1, pasa al estado de aceptación q3 y se queda ahí si sigue leyendo 1, si lee un 0, pasa al estado de rechazo q4 y así sucesivamente.