



Escuela de Ingeniería en Computación

Compiladores e Interpretes

I Semestre 2016

Apuntes Clase del 17/3/2017

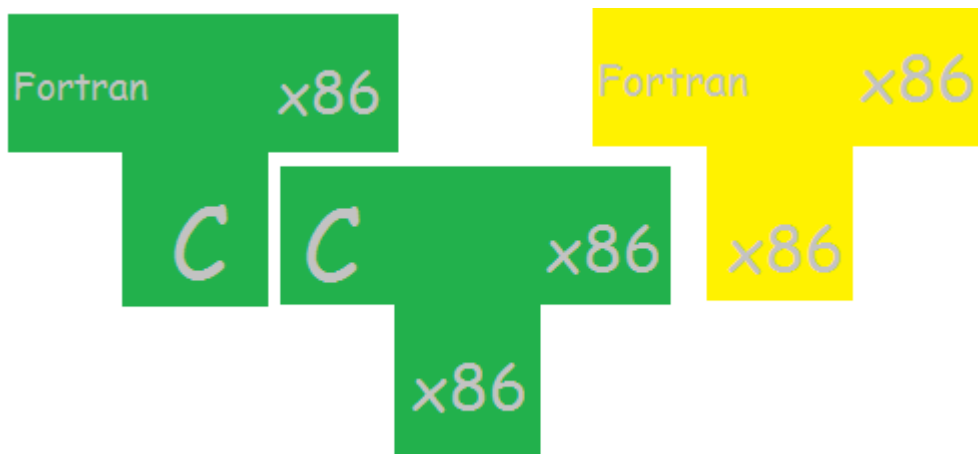
Profesor: PhD Francisco J. Torres Rojas

Apuntador: Sergio Moya Valerín (2013015682)

Tabla de contenido

Aclaración Diagramas T (Posible notación de examen)	3
Continuación Autómatas clase anterior	3
Características	5
Ejemplo 1	6
Ejemplo 2	6
Ejemplo 3	7
Implementación DFA	7
Ejemplo de código	8
Ejemplo:	8
DFA y Lenguajes	9
Nuestra misión	9
Diseño de DFA's	9
Conceptos básicos	9
Práctica en clase	10

Aclaración Diagramas T (Posible notación de examen)

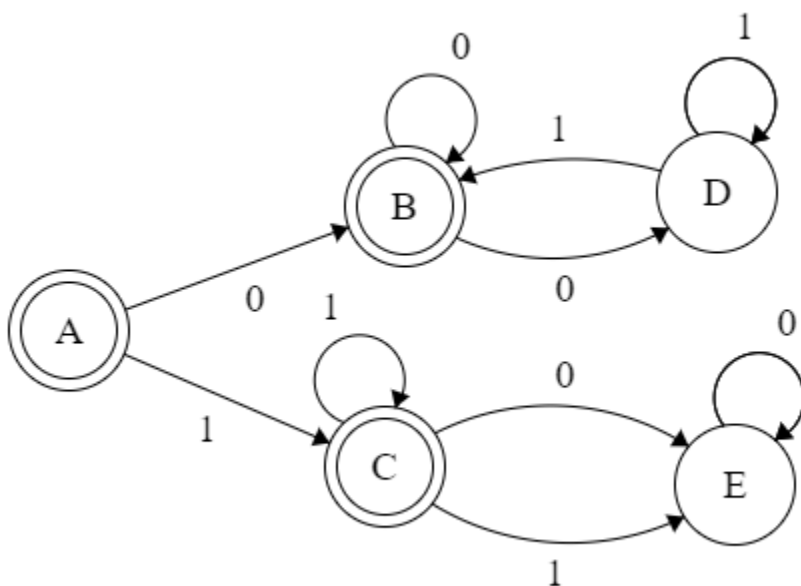


El resultado de la “ecuación” se pone arriba del segundo diagrama

Continuación Autómatas clase anterior

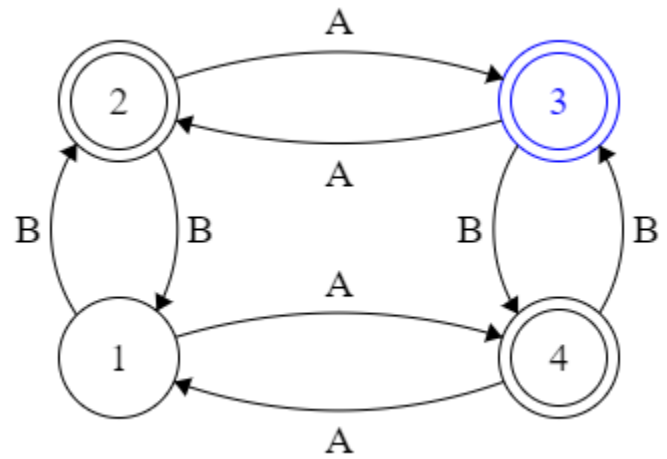
1) ¿Se acepta la hilera 000101011101?

¡Se rechaza! (La hilera debe terminar con lo que inicio)



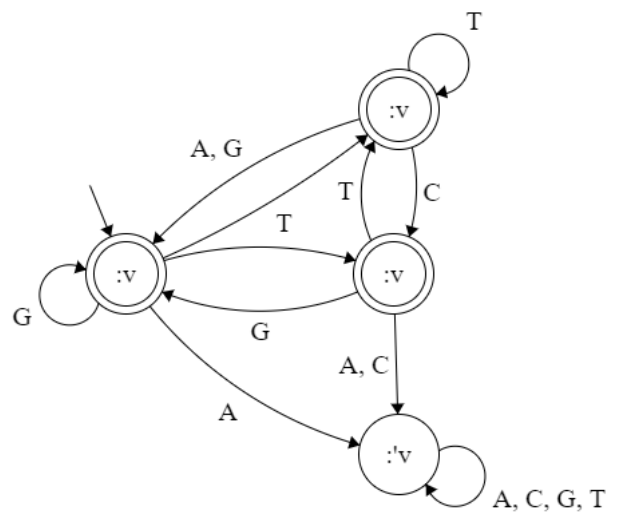
2) ¿Se acepta la hilera ϵ ?

¡Se rechaza! (EL estado inicial es de rechazo)



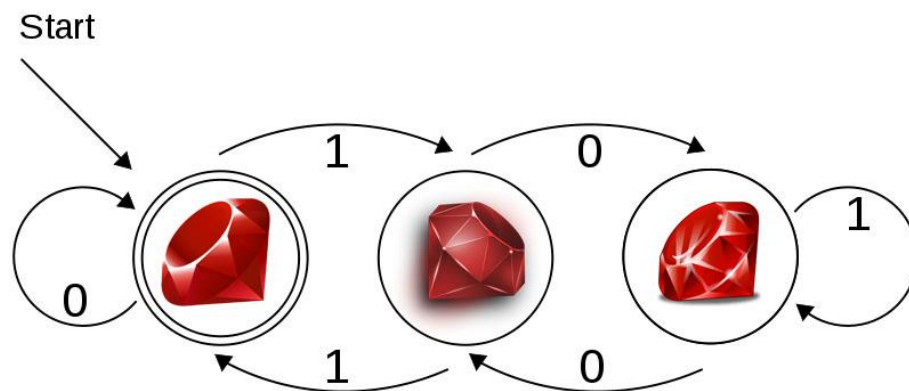
3) ¿Se acepta la hilera ϵ ?

¡Se acepta!



Características (Autómatas determinísticos de estados finitos)

- Se le dice “Maquina”
- Se le dice DFA por su significado en inglés (Deterministic Finite Automaton)
- Es finito porque hay un número fijo y finito de estados posibles
- Determinísticos
 - Dado un estado y un símbolo hay un único estado siguiente
 - Dada una entrada siempre podemos predecir adónde vamos a terminar
 - Se rechaza la hilera si no tiene para donde ir



Un autómata determinístico de estados finitos (DFA) es un quinteto

- $M = (Q, \Sigma, \delta, q_0, F)$
- Q es un conjunto finito de estados
- Σ es un alfabeto
- $\delta: Q \times \Sigma \rightarrow Q$ es la función de transición
- $q_0 \in Q$ es el estado inicial
- $F \subset Q$ Estados de aceptación (por definición podría ser vacío ¿para qué?)

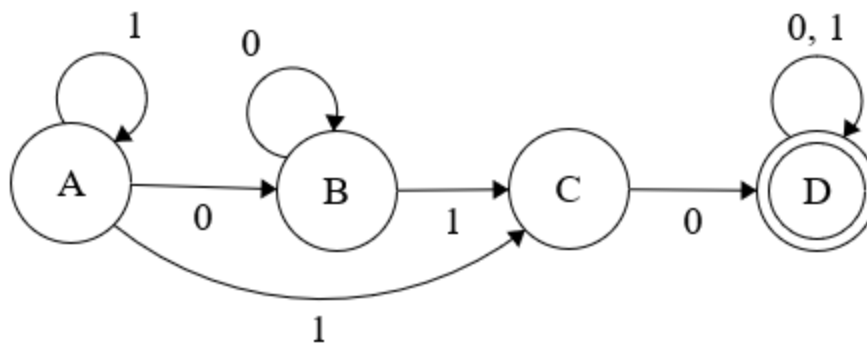
Ejemplo 1

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

Donde:

- $Q = \{a, b, c, d\}$
- $\Sigma = 1, 0$
- $q_0 = a$
- $F = \{d\}$

δ	0	1
a	b	a
b	b	c
c	d	a
d	d	d

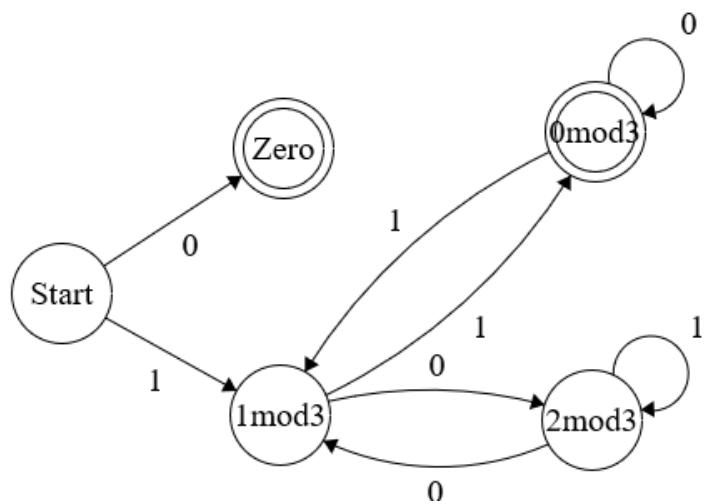


Ejemplo 2

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

Donde:

- $Q = \{\text{start}, \text{zero}, 0\text{mod}3, 1\text{mod}3, 2\text{mod}3\}$
- $\Sigma = 1, 0$
- $q_0 = \text{start}$
- $F = \{\text{zero}, 0\text{mod}3\}$



Ejemplo 3

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

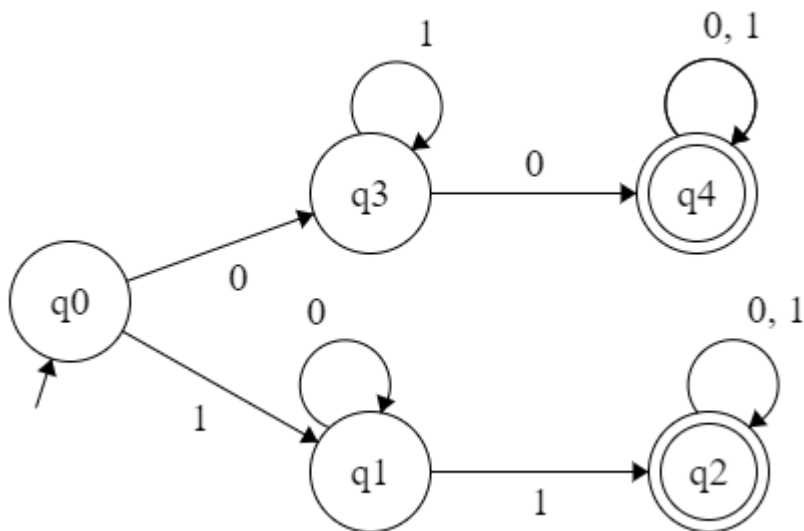
$\Sigma = \{0,1\}$

$Q = \{q_0, \dots, q_4\}$

$q_0 = q_0$

$F = \{q_2, q_4\}$

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



Nota: Si se pone Σ en un arco, significa que acepta cualquier símbolo del alfabeto

Nota: Si en algún momento “no hay a donde ir”, se da por finalizado y se rechaza la hilera (Estado de negación)

Implementación DFA

¿Cómo lo programamos?

- El algoritmo es “casi” el mismo...
- Lo que cambia son las transiciones (tabla)
-
- Tabla:
 - Filas: estados
 - Columnas: símbolo
 - Contenido: siguiente estado
- Vector de aceptado/ rechazo

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

- Se puede hacer un driver general para DFAs
- Argumentos:
 - Tabla transiciones
 - Vectores de aceptaciones
 - Puntero a función de codificación de símbolo
 - Hilera a ser revisada
 - Estado inicial
- Regresa true o false

Ejemplo de código

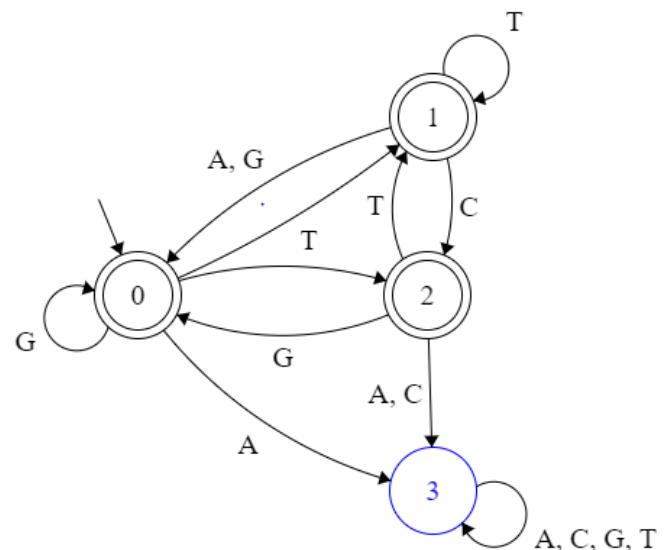


```
int DFA_driver (int **Table, int a*accept, int (*code)(char c), char* string, int state)
{
    char *s;
    int k;
    s = string;
    k = state;
    while(*s)
        k = table[k][code(*s++)]; // DEMASIADO ELEGANTE
    return (accept[k]);
}
```

Ejemplo:

Se pasarían estos parámetros al DFA_Driver y funcionaria para este diagrama.

```
int tabla[4][4] = {{3,1,2,0},{0,1,2,0},
                  {3,1,3,0},{3,3,3,3}}
int accept[4] = {1,1,1,0}
int code(char c)
{
    if(c == 'A') return 0;
    if(c == 'A') return 1;
    if(c == 'A') return 2;
    else return 3;
}
```



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, \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 $i = 0, \dots, n-1$
- $r_n \in F$

La máquina M reconoce al lenguaje L si y solo si

- $L = \{w \mid M \text{ acepta } w\}$

Nuestra misión

- Hacer un DFA que reconozca un lenguaje
- Parte del scanner de un compilador típico
- Se puede automatizar (FLEX, BISON)
- Pero debemos saber de adonde vienen

Diseño de DFA's

Nos describen un lenguaje L y debemos hacer un DFA que lo reconozca

- El DFA tiene que aceptar todas las hileras de L
- El DFA no puede aceptar hileras que no sean miembros de L

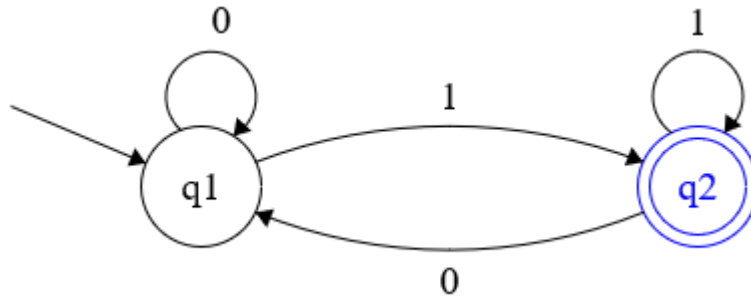
Nota: Estas condiciones son diferentes una de la otra y son indispensables

Conceptos básicos

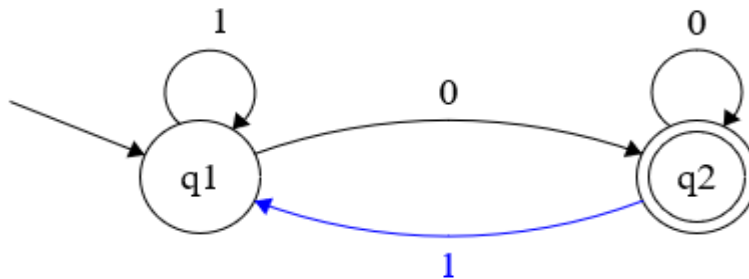
- Las etiquetas de los estados son nuestras amigas (son una ayuda)
- Numero finito de estados
- En principio de cada estado salen transiciones con todos los elementos del lenguaje
- Debemos preguntarnos:
 - ¿Cuáles estados se deben aceptar?
 - Cuestionar el primer estado, ¿ ϵ es parte del lenguaje? y con eso tenemos el primer estado
(Si el problema es de 10 estados nos ponen un 1 😊)

Práctica en clase

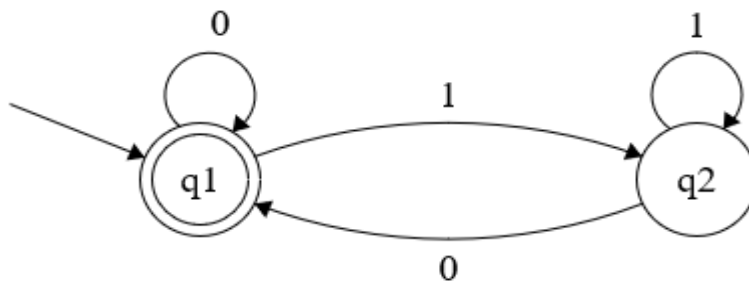
- 1) Sea L el lenguaje sobre $\Sigma = \{0,1\}$ de hileras que terminan en 1, diseñe un DFA que lo reconozca:



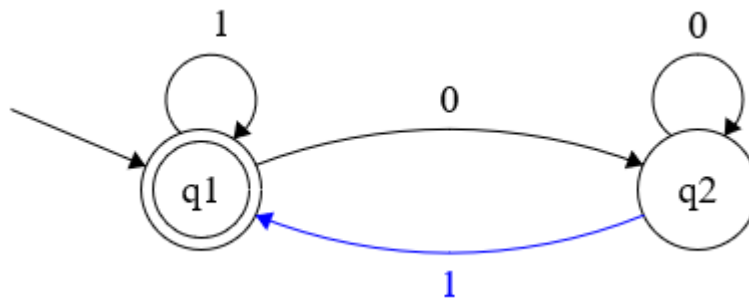
- 2) Sea L el lenguaje sobre $\Sigma = \{0,1\}$ de hileras que terminan en 0, diseñe un DFA que lo reconozca:



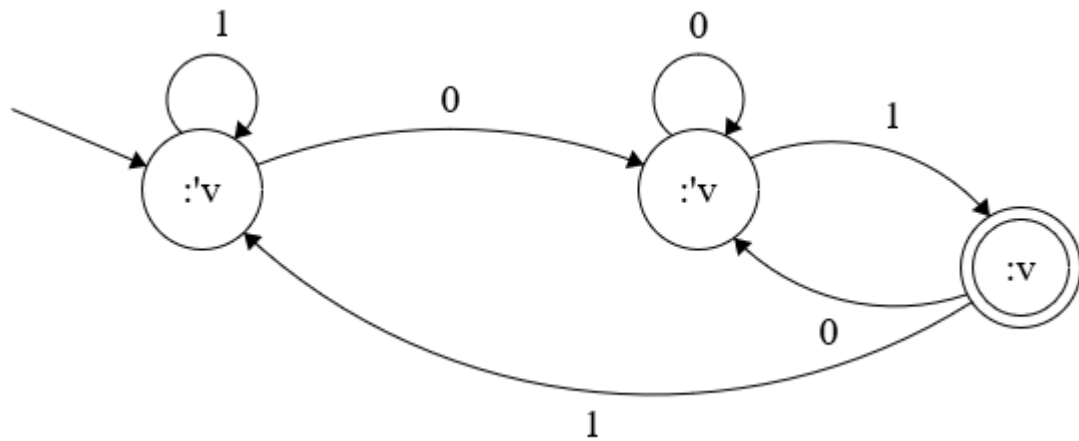
- 3) Sea L el lenguaje sobre $\Sigma = \{0,1\}$ de hileras que NO terminan en 1, diseñe un DFA que lo reconozca:



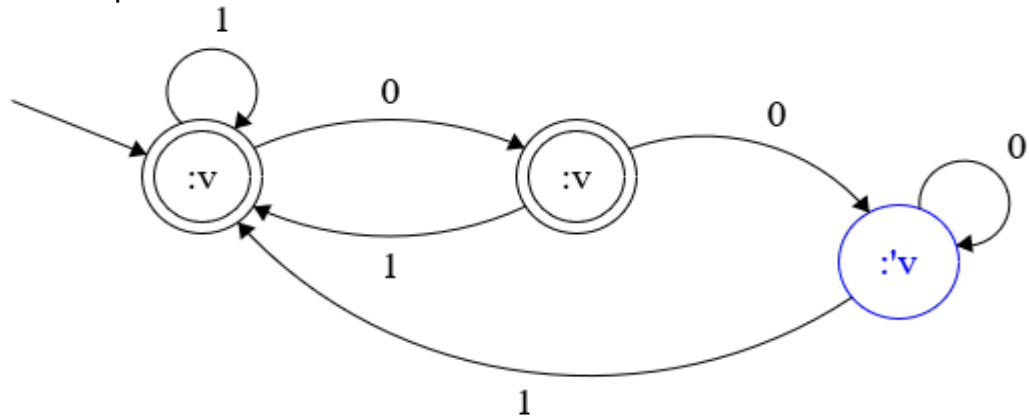
- 4) Sea L el lenguaje sobre $\Sigma = \{0,1\}$ de hileras que NO terminan en 0, diseñe un DFA que lo reconozca:



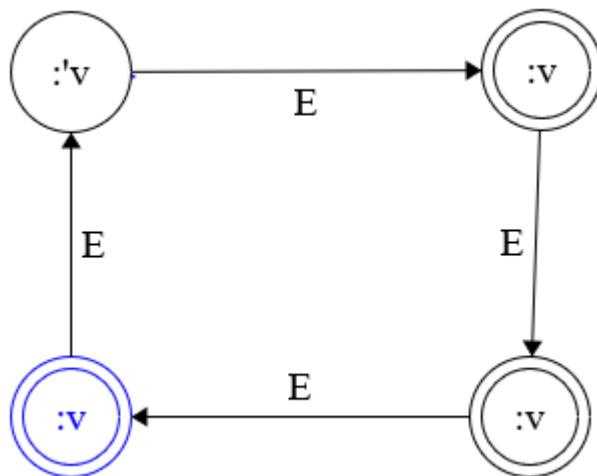
- 5) Sea L el lenguaje sobre $\Sigma = \{0,1\}$ de hileras que terminan en 01, diseñe un DFA que lo reconozca:



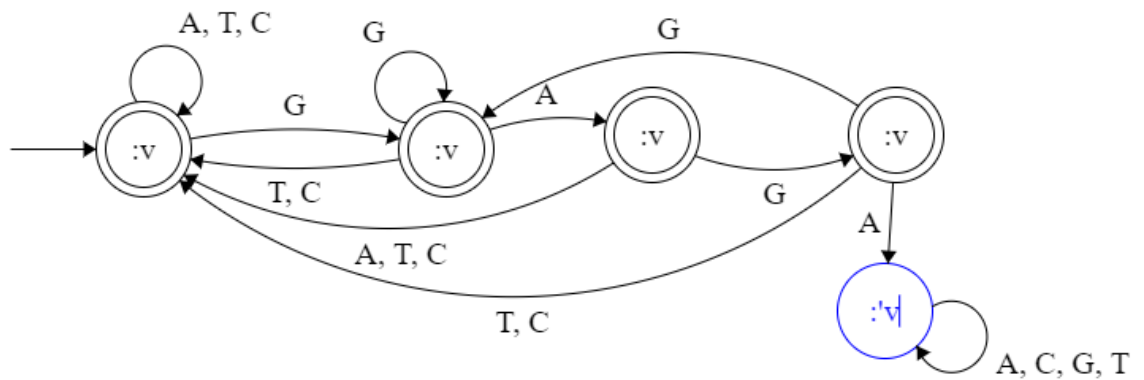
- 6) Sea L el lenguaje sobre $\Sigma = \{0,1\}$ de hileras que NO terminan en 00, diseñe un DFA que lo reconozca:



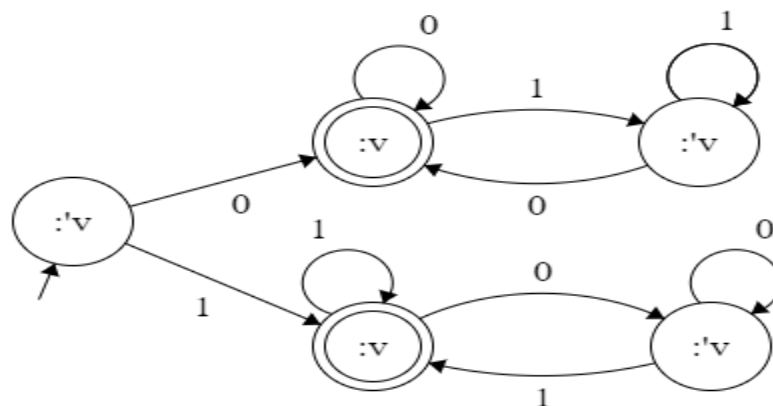
- 7) Sea L el lenguaje sobre $\Sigma = \{0,1\}$ de hileras cuya longitud no sea múltiplo de 4, diseñe un DFA que lo reconozca:



- 8) Sea L el lenguaje sobre $\Sigma = \{A, C, G, T\}$ de hileras que NO contienen la palabra GAGA, diseñe un DFA que lo reconozca:



- 9) Sea L el lenguaje sobre $\Sigma = \{0,1\}$ de hileras que terminen con el mismo símbolo que empezaron, diseñe un DFA que lo reconozca:



Nota: los diagramas fueron hechos en la página <http://madebyevan.com/fsm/>

Nota: Todas las DFA's anteriores tienen la aprobación de Chuck Norris

