

Teoría de Autómatas y Lenguajes Formales.

AFD para múltiplos de 3.

Luis José Quintana Bolaño

1 de noviembre de 2013

Resumen

Encontrar un AFD, que llamaremos A, que reconozca números binarios múltiplos de 3. Es decir:

$$L(A) = \{0, 11_{(\text{el } 3)}, 110_{(\text{el } 6)}, 1001_{(\text{el } 9)}, \dots\}$$

1. Grafo de transiciones

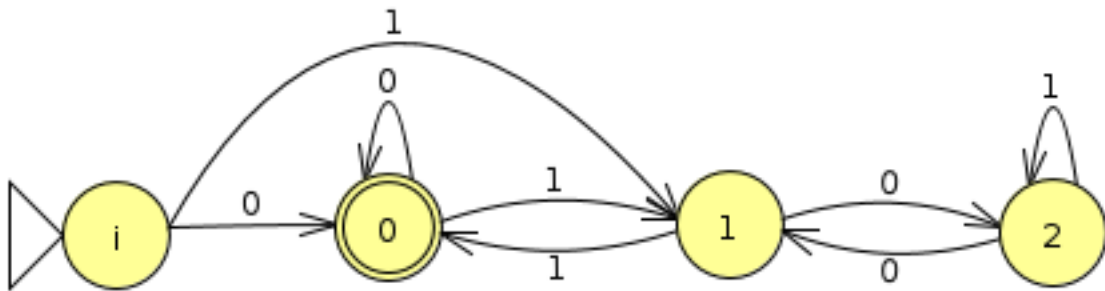


Figura 1: Grafo de transiciones

2. Tabla de transiciones

	0	1
i	0	1
*0	0	1
1	2	0
2	1	2

Figura 2: Tabla de transiciones

3. Corrección y Completitud

Demostración por inducción múltiple.

$$\text{Hipótesis de inducción} \equiv \begin{cases} \delta(i, x) = 0 & \Rightarrow & x \bmod 3 = 0 \\ \delta(i, x) = 1 & \Rightarrow & x \bmod 3 = 1 \\ \delta(i, x) = 2 & \Rightarrow & x \bmod 3 = 2 \\ \delta(i, x) = i & \Rightarrow & x \equiv \epsilon \end{cases} \quad (1)$$

Caso base: $|w| = \epsilon$

Asumimos que la hipótesis (1) es cierta para cadenas de longitud inferior a la de w , siendo $|w| \geq 1$. Como w no es vacía, podemos considerar $w = xa$, donde a es el último miembro de w y x es la cadena que la precede, por tanto la hipótesis de inducción (1) se cumple para x .

Procedemos a demostrar su corrección para $w = xa$:

1. Si $\delta(i, w) = 0$, a puede ser 0 o 1:

a) Si $a = 0$, de acuerdo a la tabla de transiciones:

- 1) $\delta(i, x) = 0$ y, por h.i., $x \bmod 3 = 0$, $x0 = x * 2$, luego $x * 2 \bmod 3 = 0$ es cierto.
- 2) $\delta(i, x) = i$ y, por h.i., $x \equiv \epsilon \rightarrow x0 = 0$, luego $0 \bmod 3 = 0$ es cierto.

b) Si $a = 1$, de acuerdo a la tabla de transiciones $\delta(i, x) = 1$ y, por h.i., $x \bmod 3 = 1 \rightarrow x = n * 3 + 1$, $x1 = 2x + 1 \rightarrow x1 = 3n + 1$, luego $3n + 1 \bmod 3 = 1$ es cierto.

2. Si $\delta(i, w) = 1$, a puede ser 0 o 1:

a) Si $a = 0$, de acuerdo con la tabla de transiciones $\delta(i, x) = 2$ y, por h.i., $x \bmod 3 = 2 \rightarrow x = n * 3 + 2$, $x0 = x * 2 \rightarrow x0 = n * 6 + 4$, luego $x0 \bmod 3 = 1$ es cierto.

b) Si $a = 1$, de acuerdo con la tabla de transiciones:

- 1) $\delta(i, x) = 0$ y, por h.i., $x \bmod 3 = 0$, $x0 = x * 2$, luego $x0 \bmod 3 = 0$ es cierto.
- 2) $\delta(i, x) = i$ y, por h.i., $x \equiv \epsilon \rightarrow x0 = 0$, luego $0 \bmod 3 = 0$ es cierto.

3. Si $\delta(i, w) = 2$, a puede ser 0 o 1:

a) Si $a = 0$, de acuerdo a la tabla de transiciones $\delta(i, x) = 1$ y, por h.i., $x \bmod 3 = 1 \rightarrow x = n * 3 + 1$, $x0 = x * 2 \rightarrow x0 = n * 6 + 2$, luego $x0 \bmod 3 = 2$ es cierto.

b) Si $a = 1$, de acuerdo a la tabla de transiciones $\delta(i, x) = 2$ y, por h.i., $x \bmod 3 = 2 \rightarrow x = n * 3 + 2$, $x1 = x * 2 + 1 \rightarrow x0 = n * 6 + 5$, luego $x1 \bmod 3 = 2$ es cierto.

La hipótesis de inducción (1) se cumple para todos los casos, por lo que el AFD es correcto y completo.

4. Implementación en C del AFD

4.1. Método de los casos

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4
5  int main(int argc, char* argv[])
6  {
7      char estado;
8      estado = 'i';
9      char c;
10     int i;
11     if(argc>1) {
12         for(i=0; i<strlen(argv[1]); i++) {
13             c=argv[1][i];

```

```

14         switch(estado) {
15             case 'i':
16                 if(c=='0') estado = '0';
17                 if(c=='1') estado = '1';
18                 break;
19             case '0':
20                 if(c=='1') estado = '1';
21                 break;
22             case '1':
23                 if(c=='0') estado = '2';
24                 if(c=='1') estado = '0';
25                 break;
26             case '2':
27                 if(c=='0') estado = '1';
28                 break;
29         }
30     }
31 }
32 printf("Estado final: %c \n",estado);
33 printf((estado=='0')?"Cadena aceptada.\n":"Cadena no aceptada.\n");
34 return 0;
35 }

```

4.2. Método de la tabla

```

1
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 int main(int argc, char* argv[])
6 {
7     int estado = 3;
8     int c;
9     int i;
10    int afd[4][2]={0,1},{2,0},{1,2},{0,1}};
11    if(argc>1){
12        for(i=0;i<strlen(argv[1]);i++) {
13            c=(argv[1][i])-'0';
14            estado=afd[estado][c];
15        }
16    }
17    printf("Estado final: %c \n", (estado==3)?'i':(estado+'0'));
18    printf((estado==0)"Cadena aceptada.\n":"Cadena no aceptada.\n");
19    return 0;
20 }

```