VERIFICACIÓN NUMERACIÓN ROMANA A TRAVÉS DE UN AUTÓMATA FINITO DETERMINISTA

Para comprobar la validez de una secuencia de numeración romana se pueden utilizar varios mecanismos. Uno muy utilizado, aplicable a una gran cantidad de situaciones diversas consiste en utilizar lo que se conoce como un **Autómata Finito Determinista (AFD)**.

Un autómata finito determinista, AFD, viene caracterizado por

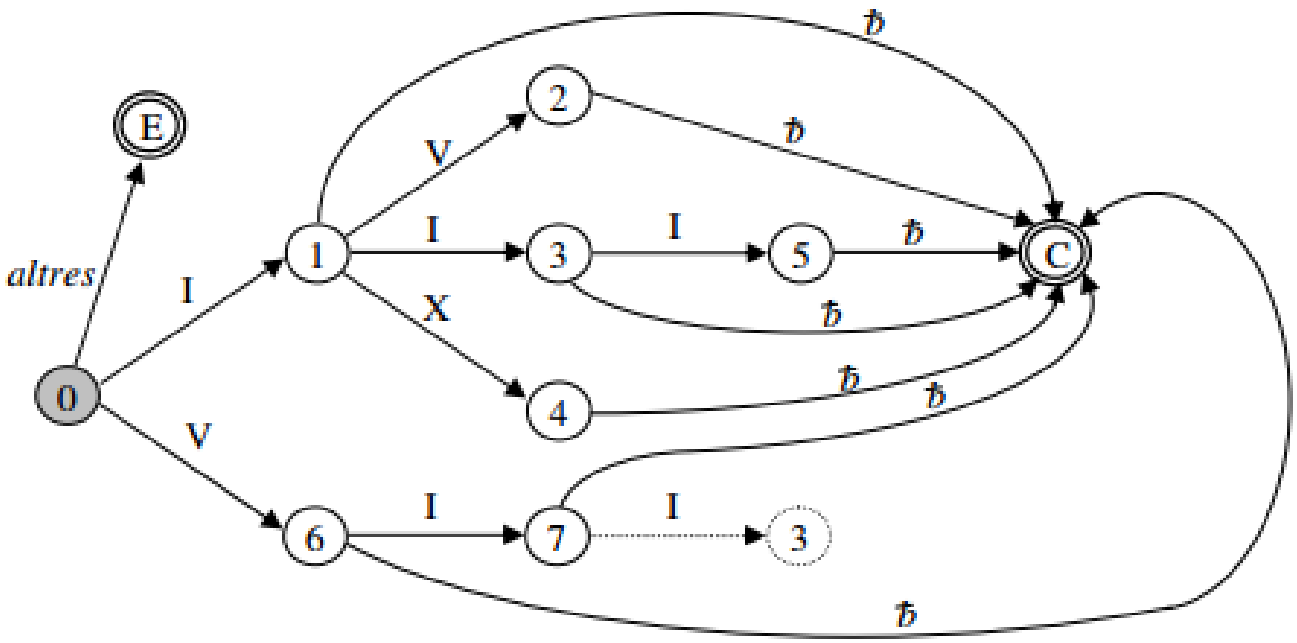
- un conjunto finito de estados Σ ,
- un conjunto finito de símbolos L,
- una función de transición de estados $T: \Sigma \times L \rightarrow \Sigma$,
- un estado $s_0 \in \Sigma$,llamado estado inicial y,
- finalmente, un conjunto de estados finales, $F \subseteq \Sigma$.

Un Autómata Finito Determinista puede representar el comportamiento que se debe tener frente a una determinada información que se suministra. Ya que dada una cierta situación (estado), y según sea la información que se reciba (símbolo de L), se pasa a otra situación diferente, según lo indique la función de transición de estados T.

VERIFICACIÓN NUMERACIÓN ROMANA A TRAVÉS DE UN AUTÓMATA FINITO DETERMINISTA

Por lo tanto un Autómata Finito Determinista es ideal para hacer de reconocedor de cifras romanas correctamente construidas. El problema de representar un AFD se resuelve utilizando un grafo, donde los nodos son los estados y las aristas que unen los nodos son las transiciones entre estados que define T en función de las entradas de símbolos de L que se puedan producir.

A continuación podemos ver un pequeño grafo que representa un AFD que reconoce cifras romanas con valor entre 1 y 9:



Se pueden ver el conjunto de estados {0, 1, 2, 3, 4, 5, 6, 7, E, C}, el estado inicial es el 0, y el conjunto de estados finales es {E, C}. también se pueden ver las transiciones que se pueden hacer desde cada uno de los estados. Es decir, se puede ver la función de transición de estado, y también se pueden ver algunos de los símbolos que se pueden reconocer. El estado inicial, 0, es el estado del que se parte. A partir del estado inicial se pueden recibir símbolos que conducirán a otro estado al autómata o bien se pueden recibir símbolos que no son correctos. En el primer caso, suponiendo que detrás del último símbolo viene un espacio en blanco, se pasa al estado C, del que no se sale, y que indica que el valor introducido es una cifra correcta. En el segundo de los casos se pasará a un estado, E, cuyo tampoco se puede salir y que indica que ha habido un error (en la grafo, por cuestiones de claridad, no se han representado todas las transiciones, en concreto las que van desde los estado 1 ... 7 hasta el estado final E no se han representado).

Utilizando este autómata se puede comprobar si la cifra introducida es o no correcta. Incluso todo puede servir para decir cuál es el valor que se ha introducido, para hacer esto sólo hay que tener en cuenta que, como en definitiva se tratará de un algoritmo, aparte de ir cambiando de estado, también se puede ir calculando el valor de la cifra que se está tratando. Si en un momento dado se determina que la cifra es errónea, entonces el valor que hasta ese momento ha sido estimando debe ser descartado.

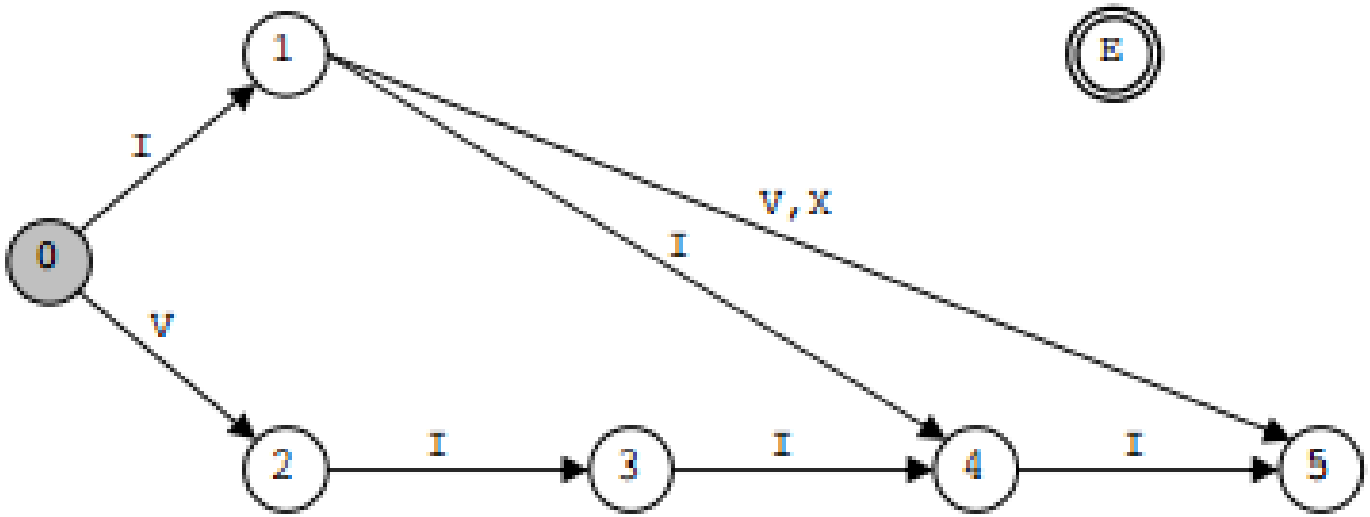
Para representar un AFD en un algoritmo se utilizan tablas para representar las transiciones entre estados. En el ejemplo, la tabla podría ser como esta:

	I	V	X	b	altres
0	1	6	E	C	E
1	3	2	4	C	E
2	E	E	E	C	E
3	5	E	E	C	E
4	E	E	E	C	E
5	E	E	E	C	E
6	7	E	E	C	E
7	3	E	E	C	E
E	E	E	E	E	E
C	C	C	C	C	C

```
char caracter;
int valor=0;
boolean final=false;
TipoEstados estado=estadoInicial;
obtenerPrimerCaracter(caracter);
while (!final) {
        estado=TABLA[estado][caracter];
        switch (estado) {
                case 1:
                case 3:
                case 5:
                case 7: valor=valor + 1;break;
                case 2: valor=5 – valor;break;
                case 4: valor=10 – valor;break;
                case 6: valor=valor + 5;break;
        }
        obtenerSiguienteCaracter(caracter);
}
if (estado==E) {
        NumeroIncorrecto;
}
else {
        NumeroCorrecto;
}
```

POSIBLE ALGORITMO PARA MANIPULAR LA TABLA ANTERIOR

El autómata visto sirve para resolver el problema planteado, pero es posible hacer lo mismo utilizando un autómata con un número de estados más reducido. Es una buena idea intentar conseguir siempre el autómata con el menor número de estados para que de esta manera las tablas implicadas sean más reducidas. Para este problema el autómata finito mínimo que reconoce los valores escritos en cifras romanas comprendidas entre I y IX es:

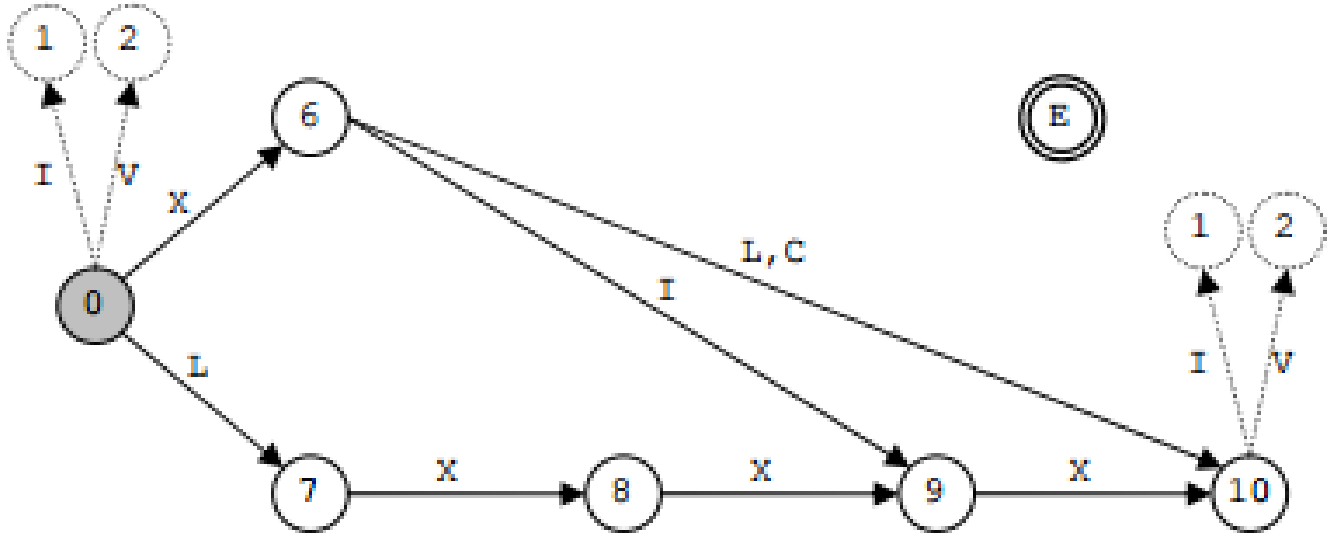


En este caso, el conjunto de estados finales es F = {1,2,3,4,5, E} de forma que al terminar, si se encuentra en el estado E, significará que lo que se ha detectado no es correcto, en cambio si lo hace en cualquier otro de los estados de F sí será un número romano correcta. La manera de terminar será encontrar un espacio en blanco o llegar al estado E ya que si se encuentra un espacio la secuencia termina y si se pasa al estado E ya no habrá manera de salir de ella. Por motivos de claridad las uniones con el estado E no se han representado.

La tabla que representa la función T sería:

	I	V	X	altres
0	1	2	E	E
1	4	5	5	E
2	3	E	E	E
3	4	E	E	E
4	5	E	E	E
5	E	E	E	E
E	E	E	E	E

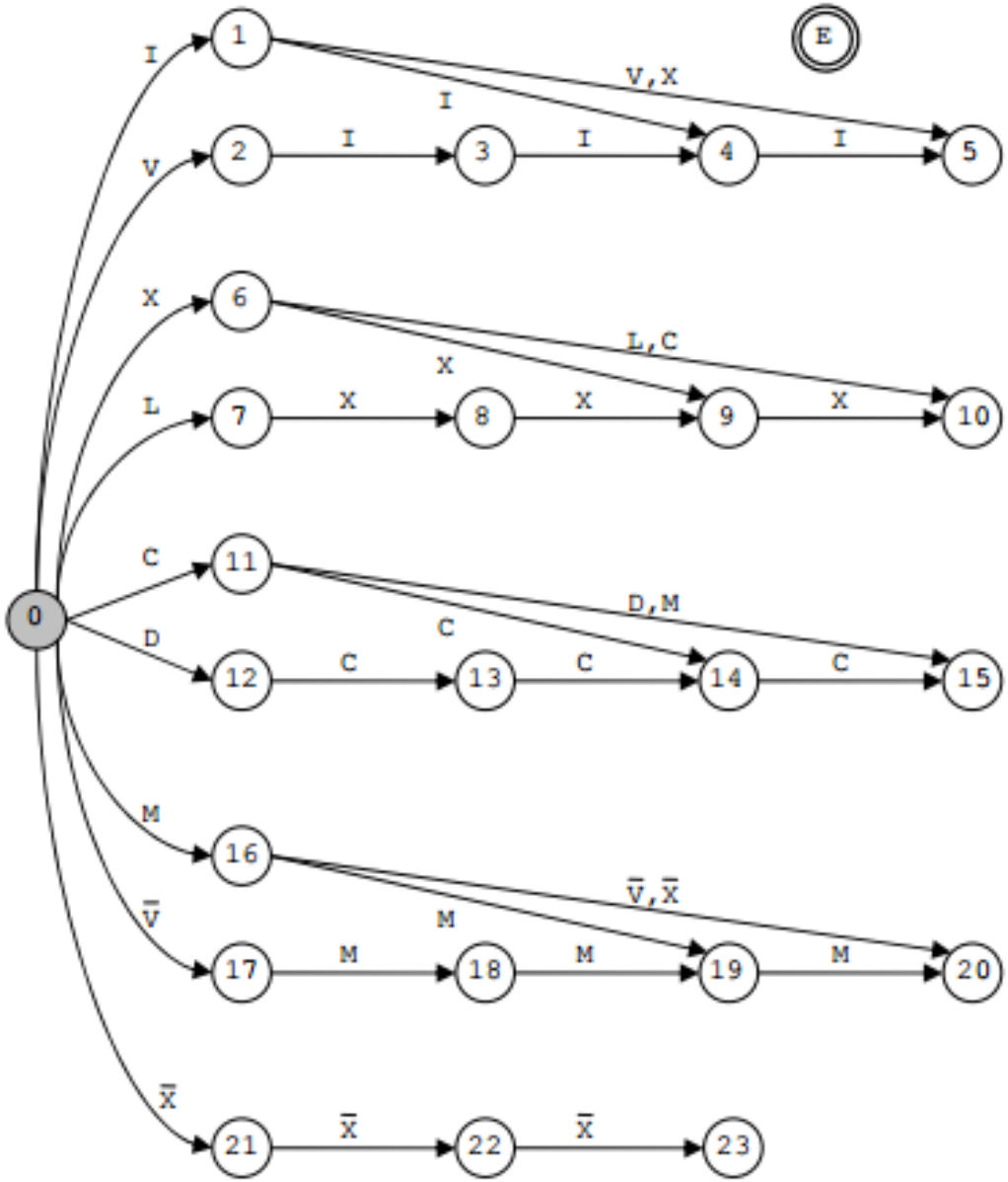
Evidentemente, tratar con un autómata todos los posibles casos que se pueden dar cuando se trata de un valor comprendido entre 1 y 39.999, se puede hacer un tanto más pesado. Pero como ya se ha visto, hay un cierto parecido en cuanto a la estructura de los diferentes valores, por ello otro pedazo del autómata completo que reconoce las decenas (los valores 10, 20, 30, 40, 50, 60, 70, 80 y 90) puede representarse de la siguiente manera:



En el grafo se repite el nodo 0, el estado inicial, y hay también repeticiones del estado final, E, así como de los nodos 1 y 2, todos ellos aparecen en el primer autómata.

A todos los nodos que corresponda deberían añadir las aristas, transiciones, necesarias para poder reconocer no sólo los valores de las decenas sino también los de las unidades, tal como se puede ver por los nodos 0 y 10, de manera que se unan los dos grafos vistos.

A continuación se puede ver el grafo que representa el autómata. No aparecen todas las aristas por motivos de claridad, pero es fácil ver de qué aristas se trata.



El Autómata Finito Determinista que reconoce los números escritos en cifras romanas con valor menor o igual a 39.999 tiene un total de 25 estados diferentes, contando el estado inicial y el estado de error. Este autómata se puede representar como una extensión de los autómatas que se han visto hasta ahora. La tabla de transiciones es la siguiente:

	I	V	X	L	C	D	M	$\overline{V}$	$\overline{X}$	altres
0	1	2	6	7	11	12	16	17	21	E
1	4	5	5	E	E	E	E	E	E	E
2	3	E	E	E	E	E	E	E	E	E
3	4	E	E	E	E	E	E	E	E	E
4	5	E	E	E	E	E	E	E	E	E
5	E	E	E	E	E	E	E	E	E	E
6	1	2	9	10	10	E	E	E	E	E
7	1	2	8	E	E	E	E	E	E	E
8	1	2	9	E	E	E	E	E	E	E
9	1	2	10	E	E	E	E	E	E	E
10	1	2	E	E	E	E	E	E	E	E
11	1	2	6	7	14	15	15	E	E	E
12	1	2	6	7	13	E	E	E	E	E
13	1	2	6	7	14	E	E	E	E	E
14	1	2	6	7	15	E	E	E	E	E
15	1	2	6	7	E	E	E	E	E	E
16	1	2	6	7	11	12	19	20	20	E
17	1	2	6	7	11	12	18	E	E	E
18	1	2	6	7	11	12	19	E	E	E
19	1	2	6	7	11	12	20	E	E	E
20	1	2	6	7	11	12	E	E	E	E
21	1	2	6	7	11	12	22	E	E	E
22	1	2	6	7	11	12	23	E	E	E
23	1	2	6	7	11	12	E	E	E	E
E	E	E	E	E	E	E	E	E	E	E