Universidad de La Laguna

E. T. S. DE INGENIERÍA INFORMÁTICA
Departamento de Estadística, I.O. y Computación
Teoría de Autómatas y Lenguajes Formales

PRACTICA 5: Autómatas Finitos Deterministas

5.1. Requisito de codificación

Cada fichero de código fuente de los que se utilicen en la práctica ha de llevar comentarios de cabecera. También es imprescindible que cada función ó método que se codifique lleve comentarios de cabecera con una breve indicación de la función que realiza.

5.2. Introducción

Un Autómata Finito Determinista (DFA) consiste en un conjunto de estados y un conjunto de transiciones entre estados que se producen a la entrada de símbolos de entrada pertenecientes a un alfabeto Σ. Para cada símbolo de entrada hay exactamente una transición para cada estado. Hay un estado especial, denominado estado inicial o de arranque que es el estado en el que el autómata se encuentra inicialmente. Por otra parte, algunos estados se denominan finales o de aceptación. Así pues, un Autómata Finito Determinista se caracteriza formalmente por una quíntupla $(\Sigma, Q, q_0, F, \delta)$ donde cada uno de estos elementos tiene el siguiente significado:

- Σ es el alfabeto de entrada del autómata. Se trata del conjunto de símbolos que el autómata acepta como entradas.
- Q es el conjunto finito de los estados del autómata. El autómata siempre se encontrará en uno de los estados de este conjunto.
- q_0 es el estado inicial o de arranque del autómata ($q_0 \in Q$). Se trata de un estado distinguido. El autómata se encuentra en este estado al comienzo de la ejecución.
- F es el conjunto de estados finales o de aceptación del autómata ($F \in Q$). Al final de una ejecución, si el estado en que se encuentra el autómata es un estado final, diremos que el autómata ha aceptado la cadena de símbolos de entrada.
- δ es la función de transición. $\delta : Q \times \Sigma \rightarrow Q$ que determina el único estado siguiente para un par (q_i, σ) correspondiente al estado actual y la entrada.

Generalmente, el término Autómata Finito Determinista lo abreviaremos AFD o más comúnmente por su acrónimo en inglés, DFA (Deterministic Finite Automaton).

Veamos un primer ejemplo de DFA definiendo los cinco elementos que lo componen:

$\Sigma = \{a, b\}$
$Q = \{q_0, q_1, q_2\}$
q_0
$F = \{q0\}$
La función de transición, δ se define por la tabla 5.1.

δ	a	b
q_0	q_1	q_2
q_1	q_2	q_0
q_2	q_2	q_2

Cuadro 5.1: La función de transición δ

En la tabla 5.1, la primera columna contiene los estados posibles del autómata, y la primera fila todos los símbolos del alfabeto. Dado un estado actual q y un símbolo de entrada x, la tabla define el estado al que transita el autómata cuando estando en el estado q, recibe la entrada x: $\delta(q, x) \in Q$

La característica esencial de un DFA es que δ es una función, por lo que debe estar definida para todos los pares del producto cartesiano $Q \times \Sigma$. Por ello, sea cual fuere el símbolo actual de la entrada y el estado en que se encuentre el autómata, siempre hay un estado siguiente y además este estado se determina de forma unívoca. Dicho de otro modo, la función de transición siempre determina unívocamente el estado al que ha de transitar el autómata dados el estado en que se encuentra y el símbolo que se tiene en la entrada.

Con un DFA siempre se puede asociar un grafo dirigido que se denomina diagrama de transición. Su construcción es como sigue: los vértices del grafo se corresponden con los estados del DFA. Si hay una transición desde el estado q hacia el estado p con la entrada i, entonces deberá haber un arco desde el nodo q hacia el nodo p etiquetado i. Los estados de aceptación se suelen representar mediante un círculo de doble trazo, y el estado de arranque se suele señalar mediante una flecha dirigida hacia el correspondiente vértice.

El diagrama de transición correspondiente al DFA de la tabla 5.1 es el que se muestra en la figura 5.1. En esa figura observamos que en este ejemplo, el estado inicial, q_0 es el único estado de aceptación del autómata (representado con el círculo de trazo doble). El estado inicial está indicado por la flecha que entra hacia él, sin partir de otro estado.

Se dice que un estado de un DFA es un estado de muerte (o de absorción) si todas las transiciones de ese estado para cualquier símbolo del alfabeto llevan a ese mismo estado. Por ejemplo, el estado 3 del DFA de la Figura 5.2 es un estado de muerte. Por convenio, a la hora de representar un autómata se supone que las transiciones de un determinado estado que no aparecen dibujadas con arcos en el diagrama de transiciones, conducen a un estado de muerte, que no se dibuja. Así el DFA de la Figura 5.2 se suele representar como aparece en la Figura 5.3, en la que el estado 3 no aparece dibujado explícitamente.

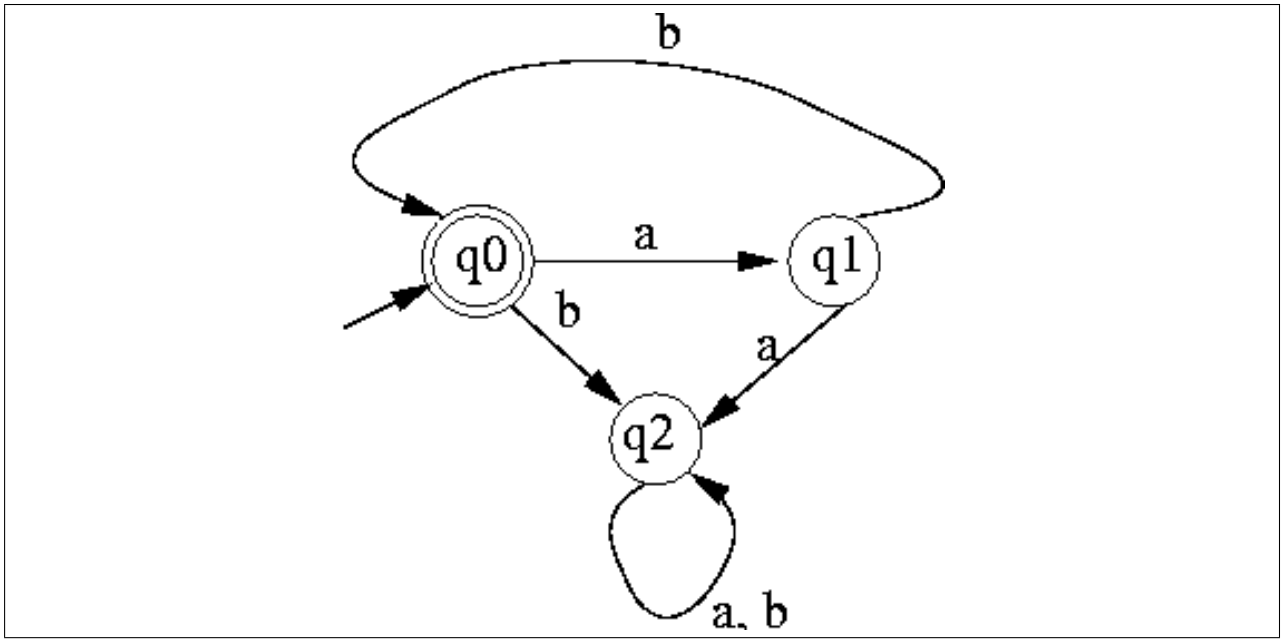

Figura 5.1: Diagrama de transiciones de un DFA

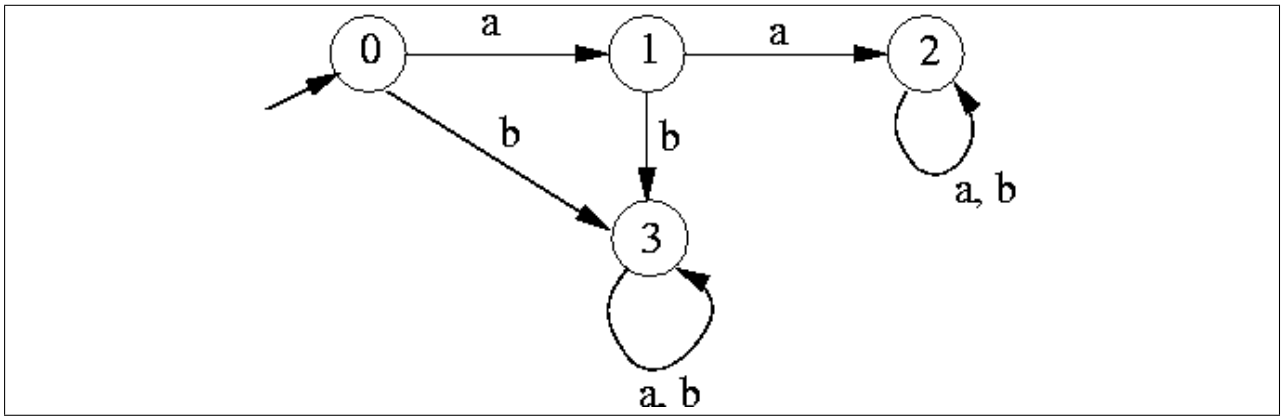

Figura 5.2: El estado 3 es un estado de muerte

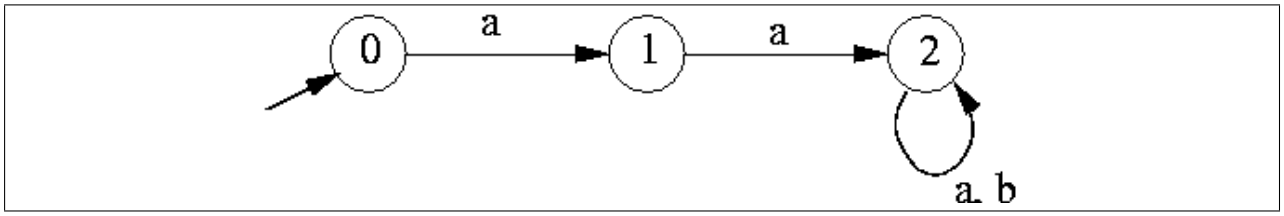

Figura 5.3: El autómata de la Figura 5.2 sin el estado de muerte

5.3. Lenguaje aceptado por un DFA

Para describir formalmente el comportamiento del autómata ante una cadena $w \in \Sigma^*$ debemos ampliar la función de transición $\delta : Q \times \Sigma \to Q$ para aplicarla sobre un estado y una cadena de símbolos en lugar de un estado y un símbolo del alfabeto. Definimos entonces $\hat{\delta} : Q \times \Sigma^* \to Q$

La idea es que $\hat{\delta}(q, w)$ será el estado que alcance el DFA después de leer la cadena w arrancando en el estado q. Dicho de otro modo, $\hat{\delta}(q, w)$ es el único estado p tal que hay un camino en el diagrama de transiciones del autómata desde q hasta p etiquetado con los símbolos de w. Formalmente definimos:

1. $\hat{\delta}(q, e) = q$
2. Para todas las cadenas $w \in \Sigma^*$ y símbolos $a \in \Sigma$: $\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$

El primer punto establece que el autómata no puede cambiar de estado sin leer ningún símbolo, y el segundo indica cómo decidir el estado en que se encuentra el autómata después de leer una cadena no vacía wa.

Dado que $\hat{\delta}(q, a) = \delta(\hat{\delta}(q, e), a) = \delta(q, a)$ las funciones δ y $\hat{\delta}$ ofrecen el mismo resultado sobre argumentos válidos para ambas (δ toma como entrada un par estado, símbolo del alfabeto mientras que $\hat{\delta}$ toma un estado y una cadena de símbolos). Por eso utilizaremos el símbolo δ para las funciones de transición de un autómata (ya sea para una cadena o un único símbolo).

Se dice que una cadena $w \in \Sigma^*$ es aceptada por un autómata $M \equiv (\Sigma, Q, q_0, F, \delta)$ si $\delta(s, w) = p$ para algún estado $p \in F$. El lenguaje aceptado por M, que se suele designar como $L(M)$ es el conjunto de cadenas $\{w | \delta(q_0, w) \in F\}$. Se dice que un lenguaje es un conjunto regular (o lenguaje regular) si es el conjunto aceptado por un DFA. Es importante aclarar que L(M) es el conjunto de todas las cadenas que el autómata acepta (no un conjunto de cadenas que acepta).

5.4. Práctica: Simulación de un Autómata Finito Determinista

La práctica consistirá en la realización de un programa en C++ que lea desde un fichero las especificaciones de un autómata finito determinista, a continuación comience a imprimir una traza de los estados por los que transita el autómata mediante las entradas en forma de caracteres que se leerán desde otro fichero, y al final indique si llegó a un estado de aceptación o no.

Tanto el nombre del fichero de definición del DFA como el de entradas para el mismo se leerán en la línea de comandos.

Todos los ficheros que se manejan en la práctica serán ficheros de texto. Los estados del autómata se representarán mediante números enteros sin signo (`unsigned`), y la numeración de los estados corresponderá a los primeros números naturales comenzando con 0.

5.4.1. Estructura de los ficheros de definición de autómatas finitos deterministas

Estos ficheros tendrán la extensión .DFA y el siguiente formato:

- Línea 1: El número total de estados del DFA.
- Línea 2: Estado de arranque del DFA.

A continuación viene una línea para cada estado, conteniendo los siguientes números, separados entre sí por espacios en blanco.

- Número identificador del estado.
- Un 1 si se trata de un estado de aceptación o un 0 si es un estado de rechazo.

A continuación (en la misma línea) para cada una de las transiciones y separados por espacios:

- Símbolo de entrada necesario para que se produzca la transición.
- Estado destino de la transición.

La Figura 5.1 presenta el fichero de definición del autómata finito determinista de la Figura 5.4.

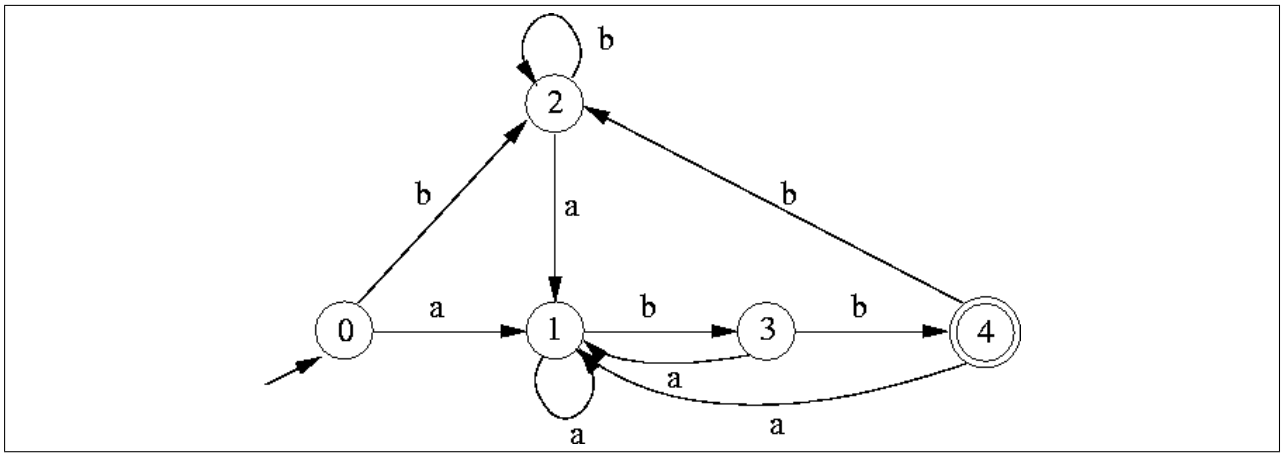

Figura 5.4: DFA que reconoce las cadenas `(a|b)*abb`

```
//
// DFA que reconoce el lenguaje representado por la expresión (a|b)*abb
//
5
0
0 0 a 1 b 2
1 0 a 1 b 3
```

```
2 0 a 1 b 2
3 0 a 1 b 4
4 1 a 1 b 2
```

Figura 5.1: Fichero de definición del DFA de la figura 5.4

La Figura 5.1 presenta el fichero de definición del autómata finito determinista de la Figura 5.4.

5.4.2. Ficheros de entradas

Estos ficheros tendrán la extensión `.in`

Se supone que el alfabeto de trabajo está constituido por todos los caracteres alfanuméricos (letras o dígitos). Las frases de entrada al autómata figuran en los ficheros de entrada sin separación entre los caracteres de la misma. Así para el autómata de la Figura 5.4 podríamos tener las entradas que se muestran en la Figura 5.2.

```
ababbbbabb
abbbbababa
```

Figura 5.2: Fichero con entradas para autómatas

5.4.3. Trazas

A medida que el autómata lea entradas deberá imprimirse una lista que contenga el número identificador del estado actual, la entrada leída y el nuevo número identificador del estado al que se transita, finalizando todo ello con un mensaje que indique si se ha aceptado la cadena de entrada o no. Así para la primera de las entradas de la Figura 5.2 obtendríamos los resultados que se presentan en la Figura 5.3.

```
Est. Actual Entrada   Nuevo Est.
    0       a         1
    1       b         3
    3       a         1
    1       b         3
    3       b         4
    4       b         2
    2       b         2
    2       a         1
    1       b         3
    3       b         4

Cadena de entrada ACEPTADA.
```

Figura 5.3: Resultado de la ejecución del programa

5.4.4. Detalles de implementación

A la hora de implementar en C++ su programa, l@s alumn@s deberán identificar los diferentes objetos que interactúan en esta simulación, y combinarlos adecuadamente para construir su programa.

No se sugerirá ninguna implementación específica para que cada alumn@ desarrolle por sí mism@ sus propias ideas. Tenga en cuenta que la simplicidad es siempre un valor añadido de todo desarrollo.

5.4.5. Comprobación de resultados

En la sección de *Descargas* de las páginas web de la asignatura, en el apartado *Autómatas Finitos* de la categoría *Prácticas* pueden encontrar algunos ejemplos de ficheros con definiciones de autómatas y de entradas (*.dfa) (*.in) con los que puede verificar su práctica.

Deberá comprobarse que la simulación tiene el comportamiento correcto, usando para ello un cierto número de autómatas, además de los suministrados.

Puede utilizar el simulador de DFAs de *ToTALF* para comprobar el comportamiento de su programa ante diferentes autómatas.