Ejemplo 3.6

Minimizar el AFD representado en el dígrafo de la Figura 3.11.

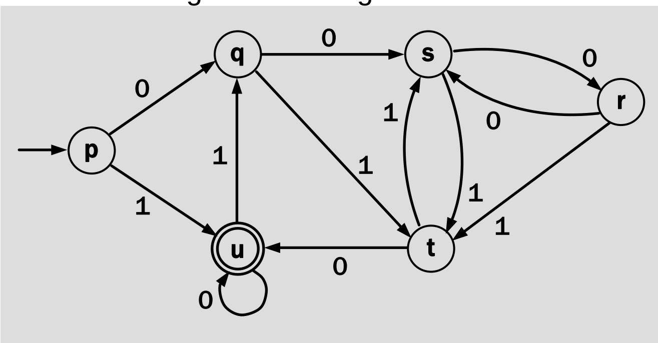

Figura 3.11: Dígrafo del AFD a ser minimizado.

La definición formal de este autómata es:

$AFD = (\Sigma_E, Q, q_0, A, f)$

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

$Q = \{p, q, r, s, t, u\}$

$q_0 = p$

$A = \{u\}$

f:	0	1
→ p	q	u
q	s	t
r	s	t
s	r	t
t	u	s
* u	u	q

Tabla 3.10: Definición y función de transición ***f*** del Ejemplo 3.6.

La primera partición del conjunto de estados separa al estado final del resto:

$$Q/E_0 = \{\{p, q, r, s, t\}, \{u\}\}$$

y los sucesivos conjuntos cocientes son identificados a partir de la función de transición, verificando el comportamiento común de los miembros de las clases de estados:

$Q/E_1 = \{\{p\}, \{q, r, s\}, \{t\}, \{u\}\}$
$Q/E_2 = \{\{p\}, \{q, r, s\}, \{t\}, \{u\}\} = Q/E_1 = Q/E = \{c_0, c_1, c_2, c_3\}$

donde $c_0 = \{p\}$, $c_1 = \{q, r, s\}$, $c_2 = \{t\}$ y $c_3 = \{u\}$

Una vez asignados los nuevos nombres a las clases de estados se representa a continuación la definición formal y grafo del AFD mínimo:

$$AFD = (\Sigma_E, Q, q_o, A, f)$$

donde $\Sigma_E = \{0, 1\}$, $Q = \{c_0, c_1, c_2, c_3\}$, $q_0 = c_0$, $A = \{ c_3 \}$.

La función de transición ***f*** se muestra en la Tabla 3.11 y el grafo en la Figura 3.12.

f:	0	1
→ c_0	c_1	c_3
c_1	c_1	c_2
c_2	c_3	c_1
* c_3	c_3	c_1

Tabla 3.11: Función de transición.

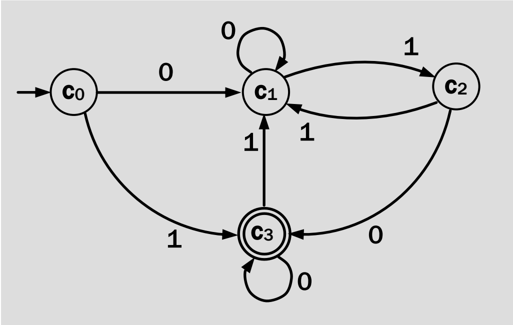

Figura 3.12: Dígrafo del AFD minimizado.

Ejemplo 3.7

En la Tabla 3.12, se presenta la función de transición de un AFD y se pide realizar un análisis completo que incluya:

a) Completar la definición formal.
b) Representar su dígrafo.
c) Verificar que el AFD sea conexo.
d) Minimizar el autómata.
e) Presentar la definición formal y grafo del AFD equivalente mínimo.
f) Identificar el lenguaje reconocido.
g) Mostrar la aceptación de una cadena de largo seis mediante el árbol de configuraciones y plano de estados-entradas.

f:	a	b
→ p_1	p_3	p_4
p_2	p_8	p_4
p_3	p_6	p_5
p_4	p_5	p_6
* p_5	p_1	p_5
p_6	p_6	p_2
p_7	p_5	p_8
p_8	p_6	p_5

Tabla 3.12: función de transición del Ejercicio 3.7

Solución

a) La definición formal queda completa con:

$$AFD = (\ \Sigma_E,\ Q,\ q_0,\ A,\ f\ )$$

donde $\Sigma_E = \{a, b\}$, $Q = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8\}$, $q_0 = p_1$ y $A = \{p_5\}$

b) Su dígrafo se presenta en la Figura 3.13:

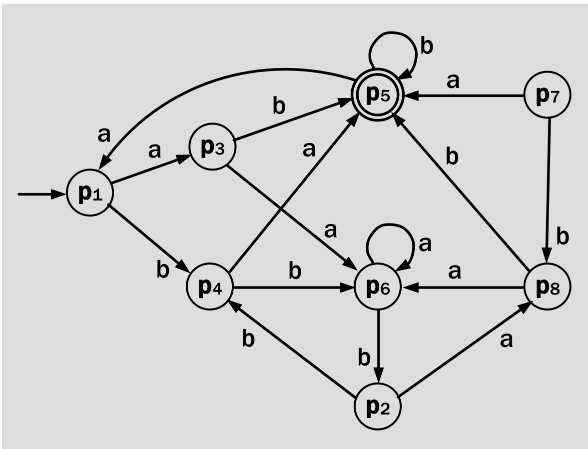

Figura 3.13: Dígrafo del AFD de la Tabla 3.12.

c) A partir de la inspección de la función de transición de la Tabla 3.12, se deduce que el AFD no es conexo, ya que el estado **p_7** no es accesible desde el estado inicial **p_1**.

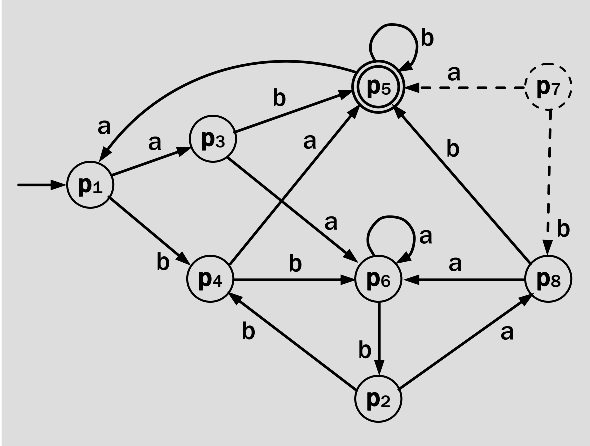

Figura 3.14: Dígrafo del AFD conexo de la Tabla 3.12.

En la Figura 3.14, se muestra con líneas de punto el estado inaccesible y las correspondientes transiciones, que son eliminadas del autómata, ya que no modifican su comportamiento.

d) Para minimizar el autómata, se identifica el conjunto cociente inicial a partir de su definición formal y luego, los sucesivos conjuntos cocientes a partir del conjunto cociente anterior y la función de transición. Se prosigue hasta obtenerse dos conjuntos cocientes sucesivos idénticos a partir del conjunto cociente inicial, que es:

$Q/E_0 = P_1^0 \cup P_2^0$, donde $P_1^0 = Q - A = \{p_1, p_2, p_3, p_4, p_6, p_8\}$ y $P_2^0 = A = \{p_5\}$

$$Q/E_0 = \{\{p_1, p_2, p_3, p_4, p_6, p_8\}, \{p_5\}\}$$

- A partir de Q/E_0 y la función de transición se comprueba que:

$f(p_1,a) = p_3 \in P_1^0$	y	$f(p_1,b) = p_4 \in P_1^0$	→	$p_1 \in P_1^1$
$f(p_2,a) = p_8 \in P_1^0$	y	$f(p_2,b) = p_4 \in P_1^0$	→	$p_2 \in P_1^1$
$f(p_3,a) = p_6 \in P_1^0$	y	$f(p_3,b) = p_5 \in P_2^0$	→	$p_3 \in P_2^1$
$f(p_4,a) = p_5 \in P_2^0$	y	$f(p_4,b) = p_6 \in P_1^0$	→	$p_4 \in P_3^1$
$f(p_5,a) = p_1 \in P_1^0$	y	$f(p_5,b) = p_5 \in P_2^0$	→	$p_5 \in P_4^1$
$f(p_6,a) = p_6 \in P_1^0$	y	$f(p_6,b) = p_2 \in P_1^0$	→	$p_6 \in P_1^1$
$f(p_8,a) = p_6 \in P_1^0$	y	$f(p_8,b) = p_5 \in P_2^0$	→	$p_8 \in P_2^1$

luego $P_1^1 = \{p_1, p_2, p_6\}$, $P_2^1 = \{p_3, p_8\}$, $P_3^1 = \{p_4\}$, $P_4^1 = \{p_5\}$ es decir que:

$$Q/E_1 = \{\{p_1, p_2, p_6\}, \{p_3, p_8\}, \{p_4\}, \{p_5\}\}$$

- Se repite el procedimiento a partir de Q/E_1 y de la función de transición:

$f(p_1,a) = p_3 \in P_2^1$	y	$f(p_1,b) = p_4 \in P_3^1$	→	$p_1 \in P_1^2$
$f(p_2,a) = p_8 \in P_2^1$	y	$f(p_2,b) = p_4 \in P_3^1$	→	$p_2 \in P_1^2$
$f(p_3,a) = p_6 \in P_1^1$	y	$f(p_3,b) = p_5 \in P_4^1$	→	$p_3 \in P_2^2$
$f(p_4,a) = p_5 \in P_4^1$	y	$f(p_4,b) = p_6 \in P_1^1$	→	$p_4 \in P_3^2$
$f(p_5,a) = p_1 \in P_1^1$	y	$f(p_5,b) = p_5 \in P_4^1$	→	$p_5 \in P_4^2$
$f(p_6,a) = p_6 \in P_1^1$	y	$f(p_6,b) = p_2 \in P_1^1$	→	$p_6 \in P_5^2$
$f(p_8,a) = p_6 \in P_1^1$	y	$f(p_8,b) = p_5 \in P_4^1$	→	$p_8 \in P_2^2$

luego $P_1^2 = \{p_1, p_2\}$, $P_2^2 = \{p_3, p_8\}$, $P_3^2 = \{p_4\}$, $P_4^2 = \{p_5\}$, $P_5^2 = \{p_6\}$ es decir que:

$$Q/E_2 = \{\{p_1, p_2\}, \{p_3, p_8\}, \{p_4\}, \{p_5\}, \{p_6\}\}$$

- Se repite el procedimiento a partir de Q/E_2 y de la función de transición:

$f(p_1,a) = p_3 \in P_2^2$	y	$f(p_1,b) = p_4 \in P_3^2$	→	$p_1 \in P_1^3$
$f(p_2,a) = p_8 \in P_2^2$	y	$f(p_2,b) = p_4 \in P_3^2$	→	$p_2 \in P_1^3$
$f(p_3,a) = p_6 \in P_5^2$	y	$f(p_3,b) = p_5 \in P_4^2$	→	$p_3 \in P_2^3$
$f(p_4,a) = p_5 \in P_4^2$	y	$f(p_4,b) = p_6 \in P_5^2$	→	$p_4 \in P_3^3$

$f(p_5,a) = p_1 \in P_1^2$ y $f(p_5,b) = p_5 \in P_4^2$ → $p_5 \in P_4^3$
$f(p_6,a) = p_6 \in P_5^2$ y $f(p_6,b) = p_2 \in P_1^2$ → $p_6 \in P_5^3$
$f(p_8,a) = p_6 \in P_5^2$ y $f(p_8,b) = p_5 \in P_4^2$ → $p_8 \in P_2^3$

luego $P_1^3 = \{p_1, p_2\}$, $P_2^3 = \{p_3, p_8\}$, $P_3^3 = \{p_4\}$, $P_4^3 = \{p_5\}$, $P_5^3 = \{p_6\}$ es decir que:

$$Q/E_3 = \{\{p_1, p_2\}, \{p_3, p_8\}, \{p_4\}, \{p_5\}, \{p_6\}\} = Q/E_2 = Q/E$$

Para completar el proceso, se asignan luego nuevos nombres a los conjuntos de estados indistinguibles:

$$c_0 = \{p_1, p_2\},\ c_1 = \{p_3, p_8\},\ c_2 = \{p_4\},\ c_3 = \{p_5\},\ c_4 = \{p_6\}$$

Se deja al lector comprobar que el mismo conjunto cociente puede también ser obtenido con el procedimiento de los esquemas de agrupamiento de estados utilizado en el Ejemplo 3.5.

e) La definición formal y dígrafo del AFD equivalente mínimo se presentan a continuación en la Tabla 3.13 y Figura 3.15:

AFD_{min} = (Σ_E, Q, q_0, A, f)
$\Sigma_E = \{a, b\}$
$Q = \{c_0, c_1, c_2, c_3, c_4\}$
$q_0 = c_0$
$A = \{c_3\}$

f:	a	b
→ c_0	c_1	c_2
c_1	c_4	c_3
c_2	c_3	c_4
* c_3	c_0	c_3
c_4	c_4	c_0

Tabla 3.13: Definición formal del AFD mínimo y función de transición.

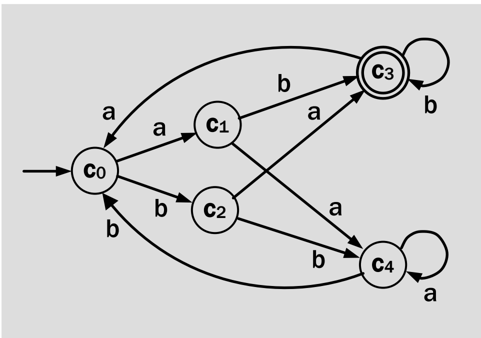

Figura 3.15: Dígrafo del AFD mínimo.

f) Para identificar el lenguaje que es reconocido por un AFD hay dos opciones: i) determinar las reglas de producción de la gramática a partir de la función de transición del autómata y ii) inducir el lenguaje a partir de inspeccionar el comportamiento del AFD. La primera opción es la más recomendable por tratarse de un proceso sistemático, y una vez identificadas las reglas de producción se deducen las formas que toman las sentencias del lenguaje. Como este tema será presentado en el próximo capítulo, por el momento se utilizará la segunda opción, que implica un proceso de inspección que, en algunos casos, puede ser muy laborioso y es susceptible a errores y omisiones. Para facilitar la tarea, es siempre recomendable trabajar sobre el AFD ya minimizado.

Estudiando el comportamiento del autómata, puede deducirse que reconoce cadenas que terminen con los sufijos abb^p o bab^p (para $p \geq 0$) y que inicien con cualquier concatenación de cadenas que lo restituyan al estado inicial:

$$\beta_1 = (abb^{n1}a)^{m1} \quad \beta_2 = (bab^{n2}a)^{m2} \quad \beta_3 = (aaa^{n3}b)^{m3} \quad \beta_4 = (bba^{n4}b)^{m4}$$

donde n_i, $m_j \geq 0$ para i, j=1, 2, 3, 4, con los mismos sufijos.

Es así que a partir de la identificación de estas cadenas se puede hacer una descripción algebraica del lenguaje que es reconocido por el AFD, que aquí es llamado M y toma la siguiente forma:

$$L(M) = \{\alpha \ / \ \alpha = ((\beta_1\beta_2\beta_3\beta_4)^i\, abb^j)^k((\beta_1\beta_2\beta_3\beta_4)^l\, bab^m)^n \text{ ó }$$
$$((\beta_1\beta_2\beta_3\beta_4)^i\, bab^j)^k((\beta_1\beta_2\beta_3\beta_4)^l\, abb^m)^n, \text{ con } i, j, k, l, m \geq 0;\ n \geq 1\}$$

Como ya fue anticipado, en el próximo capítulo, se presentará un procedimiento sistemático para deducir las reglas de producción de la gramática que genera el lenguaje aceptado por un AFD y que permitirá confirmar las formas generales propuestas, lo que se deja reservado al lector.

g) Para mostrar el comportamiento del autómata de selecciona una cadena tal como δ=bbbabb$\in$L(M), que responde a la forma general β4abbp. Con el fin de seguir su desempeño en el proceso de aceptación de la cadena, se utilizan las representaciones del árbol de configuraciones o descripciones instantáneas y el plano estados-entradas, que son mostrados en las Figuras 3.16 y 3.17.

$(c_0$, bbbabb$)$ Configuración inicial
|
$(c_2$, bbabb$)$
|
$(c_4$, babb$)$
|
$(c_0$, abb$)$
|
$(c_1$, bb$)$
|
$(c_3$, b$)$
|
(c_3, λ) Configuración final

Figura 3.16: Árbol de descripciones instantáneas.

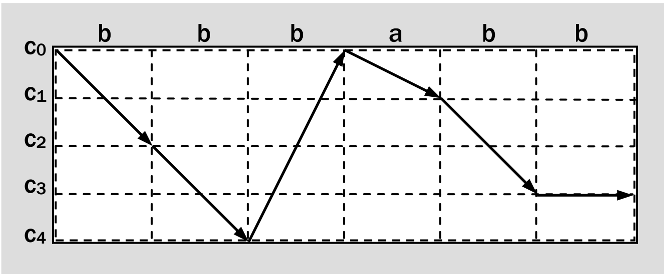

Figura 3.17: Representación del plano estados-entradas.

Ejemplo 3.8

Como se señaló en el Capítulo 1, la construcción de modelos es una actividad fundamental en el desarrollo de software y, para ello, suelen utilizarse máquinas de estados. En este caso, se propone la definición del autómata finito que represente el funcionamiento de un reproductor de audio; en la actualidad de música almacenada en tarjetas de memoria, CD o DVD y, anteriormente, sobre cinta magnética. Debe notarse que cualquiera sea la tecnología de almacenamiento, el problema sigue siendo el mismo: reconocer los comandos ingresados al reproductor por el usuario. Además, debe también reconocerse que se ha alcanzado el extremo inicial o final de la información almacenada, en este caso, música (supondremos funcionamiento circular).

En necesario relacionar las señales externas a ser reconocidas por el autómata con el significado de las mismas en el contexto del equipo representado y lo mismo ocurre con los estados previstos, donde cada uno de ellos representa una condición de operación. Con este fin, las señales de entrada son definidas en la Tabla 3.14 y los estados previstos son definidos en la Tabla 3.15:

Comandos reconocidos	
símbolo	significado
a	PLAY
b	FORWARD
c	STOP
d	PAUSA
e	REWIND
f	Marca de inicio/fin

Tabla 3.14: Señales reconocidas.

Condiciones del equipo	
símbolo	significado
p	OFF
q	ON
r	RETROCEDE
s	AVANZA
t	PAUSA

Tabla 3.15: Estados propuestos.

a) Definición formal del autómata finito:

$$
\begin{aligned}
&AFD = (\Sigma_E, Q, q_0, A, f) \\
&\Sigma_E = \{a, b, c, d, e, f\} \\
&Q = \{p, q, r, s, t\} \\
&q_0 = p \\
&A = \{p\}
\end{aligned}
$$

Donde la función de transición *f* es presentada en la Tabla 3.16:

f:	a	b	c	d	e	f
→p *	q	s	p	p	r	p
q	q	s	p	t	r	q
r	q	s	p	r	r	r
s	q	s	p	s	r	s
t	t	t	t	q	t	t

Tabla 3.16: Función de transición del AF.

b) Dígrafo: A partir de la definición formal anterior se desarrolla el dígrafo del autómata finito que cumple la función de un reproductor de audio, el cual se muestra en la Figura 3.18. Nótese que, tal como fue definida esta máquina, el estado inicial es también su estado de aceptación.

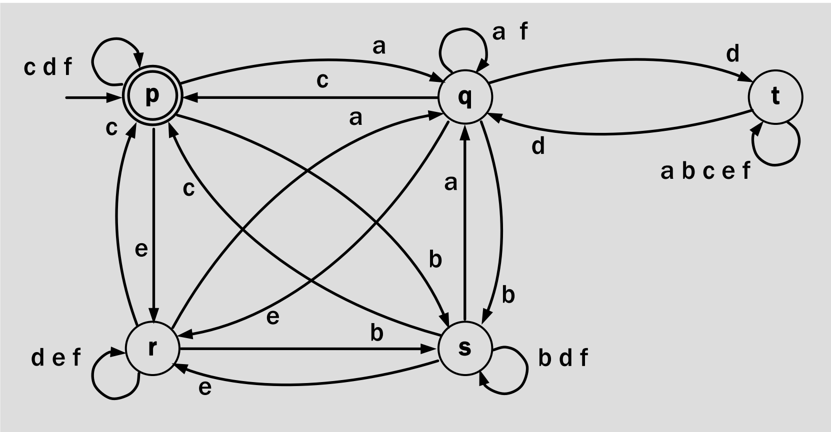

Figura 3.18: AF para un reproductor de audio.

c) Minimización del autómata finito: el objeto es verificar la posible existencia de estados indistinguibles. Se identifica el conjunto cociente inicial y luego los sucesivos conjuntos cocientes hasta alcanzar dos conjuntos cocientes sucesivos idénticos. Estos resultan ser:

$Q/E_0 = \{\{q, r, s, t\}, \{p\}\}$
$Q/E_1 = \{\{q, r, s\}, \{t\}, \{p\}\}$
$Q/E_2 = \{\{q\}, \{r, s\}, \{t\}, \{p\}\}$
$Q/E_3 = \{\{q\}, \{r, s\}, \{t\}, \{p\}\}$

El proceso de minimización muestra que los estados r y s son equivalentes o indistinguibles, ya que tienen idéntico comportamiento en cuanto a sus transiciones desde y hacia los demás estados. Esto es así en el autómata finito del modelo. Sin embargo, en el equipo reproductor de audio, el estado r corresponde a la acción de retroceder sobre el medio de entrada, mientras que el estado s está asociado a un movimiento en sentido opuesto, es decir, adelantar sobre el mismo medio. Esta aparente inconsistencia no es tal, ya que al suponer el funcionamiento circular, al avanzar más allá del último tema de música se continúa con el primero y al retroceder, en el primero, se continúa con el último.

Ejemplo 3.9

Una de las principales aplicaciones de los autómatas finitos es la identificación de los componentes del texto de un documento, pudiendo estar inserto ya sea en un corrector ortográfico o en un compilador, entre otros. En este último caso, el autómata finito es denominado *analizador léxico* (en realidad, constituye la rutina de reconocimiento del módulo denominado con ese nombre) y dentro de un compilador su misión es identificar las palabras reservadas, variables, constantes, operadores y demás elementos que forman las sentencias del programa.

Debe tenerse presente que el analizador léxico recibe el programa fuente como una larga cadena de caracteres, con espacios y caracteres especiales de control intercalados (<CR>, <LF>, <TAB>, etcétera) y debe separar e identificar los componentes del lenguaje, llamados *componentes léxicos* o *tokens*. Las posibles secuencias de caracteres que definen un *token* son verificadas, ya que deben responder a ciertas expresiones regulares, y clasificadas. Luego se les asigna una dirección en la tabla de símbolos y sirven de base a la siguiente etapa del proceso de compilación, que es denominada de análisis sintáctico, tal como se describe en detalle en el Apéndice A.

En este ejemplo, se desea construir un analizador léxico que, en un programa escrito en lenguaje Java, identifique los valores numéricos (constantes) en sus diferentes tipos (int, float, double) y notaciones, cuya representación es resumida en la Tabla 3.17, mostrada a continuación.

Tipo	Notación	Representación	Caracter (n)
Entero	Decimal	[±] nnn…n	0…9
	Octal	[±] 0nnn…n	0…7
	Hexadecimal	[±] 0xnnn…n	0…F
Real	Dec. s/ exp.	[±] n…n.n…n	0…9
	Dec. c/ exp.	[±] n.n…nE[±]nn	0…9

Tabla 3.17: Representación de constantes numéricas en Java.

Se quiere entonces determinar: a) el grafo del autómata finito que cumple la función de analizador léxico y b) su definición formal.

a) Definición de la simbología y definición formal

Para resolver el problema, se propone un AF que opere los símbolos especiales definidos en la Tabla 3.18 y con tantos estados de aceptación como las diferentes formas numéricas que deben ser identificadas, que en este caso son cinco (tres formas enteras y dos reales, Tabla 3.19).

Simbología especial	
símbolo	representa
b	blanco, <cr>,<lf>,<tab>
;	fin de sentencia
.	punto decimal
+ , -	signos de valor numérico
β	≠ 0..9 , + , -

Tabla 3.18: Señales reconocidas.

Estados de aceptación	
símbolo	**significado**
c_0	entero decimal
c_1	real sin exponente
c_2	real con exponente
c_3	entero hexadecimal
c_4	entero octal

Tabla 3.19: Estados de aceptación.

Definición formal:

$AFD = (\sum_E, Q, q_0, A, f)$,

donde:

$\sum_E = \{ 0, ..., F, +, -, b, ;, ., E, x \}$

$Q = \{ p, q, r, s, t, u, v, w, x, y, z, c_0, c_1, c_2, c_3, c_4\}$

$q_0 = p$

$A = \{c_0, c_1, c_2, c_3, c_4\}$

f:	0	1..7	8..9	A..F	+,-	x	.	E	b,;	β
->p	r	t	t		q					p
q	r	t	t							
r		z				y	s		c_0	
s	u	u	u							
t	t	t	t						c_0	
u	u	u	u					v	c_1	
v	x	x	x		w					
w	x	x	x							
x	x	x	x						c_2	
y	y	y	y	y					c_3	
z	z	z							c_4	
*c_0										
*c_1										
*c_2										
*c_3										
*c_4										

Tabla 3.20: Función de transición del analizador léxico.

Nótese que el AFD llegará a uno de los estados finales cuando complete la identificación de un componente numérico del código del programa. El estado alcanzado dependerá del tipo y notación del elemento encontrado. Luego el AFD debe regresar al estado inicial para esperar la llegada de la próxima cadena y continuar operando hasta completar el análisis léxico, destinado a identificar los números presentes en todas las sentencias del programa.

También es necesario destacar que el símbolo β es genérico, representando cualquier carácter inválido como inicio de una cadena que contenga un número. Es decir que ante la lectura de cualquier símbolo $a \notin \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, - \}$ el autómata permanecerá en el estado inicial “p”.

b) Grafo del AFD (analizador léxico)

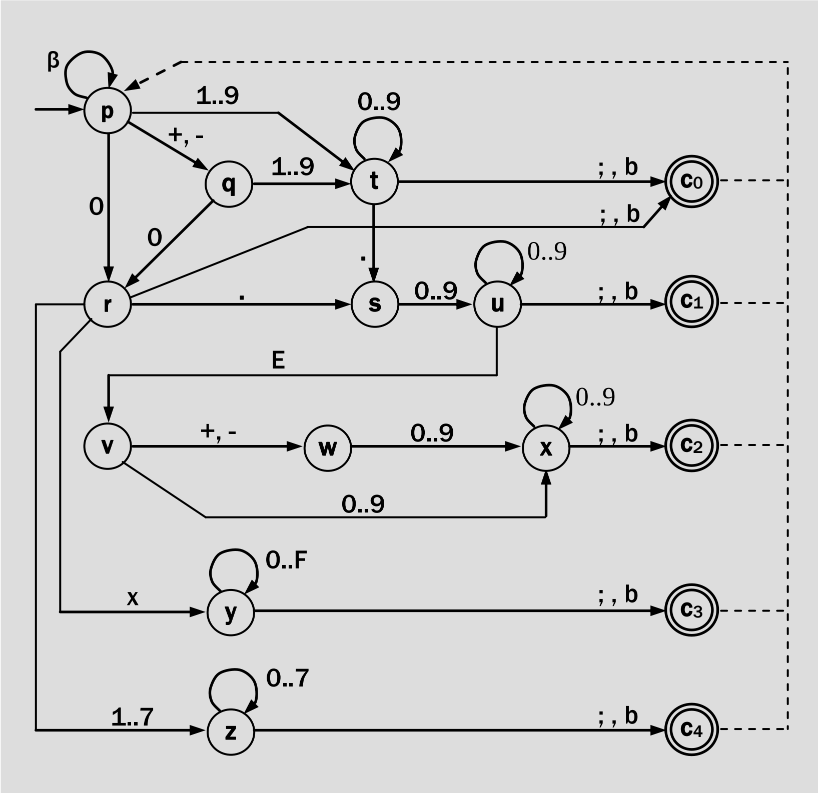

Figura 3.19: Autómata finito para un analizador léxico.

Autómatas finitos bidireccionales

Aspectos generales

Los autómatas finitos leen siempre la cadena de entrada de izquierda a derecha. Esto tiene varias consecuencias: *i*) cada símbolo de la cadena es leído una única vez, *ii*) al completarse la lectura, la cadena es aceptada o no, según se alcance un estado de aceptación, *iii*) puede anticiparse la cantidad de intervalos de tiempo necesarios para evaluar la cadena, que será igual a su largo y *iv*) en todo momento está claramente definida la subcadena pendiente de ser leída por el autómata. Esto último soporta el concepto de configuración o descripción instantánea del AFD.

Otorgando a un AFD la capacidad de decidir el sentido del movimiento del cabezal en cada intervalo de tiempo queda conceptualmente definido el AFDB (*Autómata Finito Determinista Bidireccional*). Como es de esperar, en la definición del autómata se debe incorporar una *función de movimiento*, que defina el sentido del movimiento del cabezal en cada estado y ante cada símbolo de entrada. Además, debe quedar claramente establecido el sector de la cinta en el que el autómata puede operar. Para ello, se incorporan dos símbolos especiales que identifican el inicio y el final del intervalo de cinta de entrada a ser analizado.

Con esto, se obtiene un autómata que cambia las condiciones antes señaladas, a saber: *i)* Cada símbolo de entrada pueden ser leído varias veces, *ii*) no hay ninguna condición que indique el fin de la lectura de la cadena, *iii*) no es posible anticipar la cantidad de intervalos de tiempo requeridos para evaluar la cadena y *iv*) el concepto de *cadena que resta de ser leída* desaparece. Se suma aquí una consecuencia novedosa, que es la real posibilidad de que ciertas cadenas de entrada lleven al AFDB a un ciclo cerrado que no conduzca a nada, es decir, un ciclo infinito.

Los AFDB son escasamente considerados en la literatura, a pesar de que presentan una forma de introducir gradualmente la máquina de Turing y de que por sí mismos dan lugar a soluciones muy interesantes. Hubo dos trabajos, debidos a Rubin y Scout (1959) y Shepherdson (1959), que a través de enfoques diferentes se ocuparon de demostrar la equivalencia de los AFD y AFDB. En la actualidad, los AFDB son principalmente utilizados en el análisis de la complejidad de algoritmos.

Definición del AFDB

A partir de lo ya expuesto, el AFDB se define como:

$$AFDB = (\Sigma_E, \Gamma, Q, q_0, A, f)$$

donde:

Σ_E **:** Alfabeto de símbolos de entrada
Γ **:** Alfabeto de cinta, $\Gamma = \Sigma_E \cup \{\vdash, \dashv\}$
Q **:** Conjunto finito, no vacío, de estados posibles.
q_0 **:** Estado inicial de operación, $q_0 \in Q$
A **:** Conjunto de estados de aceptación, $A \subseteq Q$
f **:** Función de transición, f: $Q \times \Gamma \rightarrow Q \times \{I, N, D\}$

Sobre la definición del AFDB nótese que:

a) El alfabeto de cinta Γ se forma con el alfabeto de entrada y dos símbolos auxiliares destinados a marcar el inicio y fin de la cinta de entrada, normalmente denominados BOT (Beginning Of Tape, $\vdash$) y EOT (End Of Tape, $\dashv$); estos símbolos nunca son de entrada.

b) El AFDB tiene normalmente un único estado de aceptación, ya que al poder ser releída la cadena siempre podrá ser definido con uno solo de estos estados, lo que no es obvio ni es demostrado aquí.

c) La función de transición es definida usando el alfabeto ampliado Γ, ya que el autómata debe reconocer todos los símbolos que pueda contener la cinta de entrada y tomar decisiones a partir de ellos.

d) La función de transición ***f*** reúne, tal como está definida, dos funciones claramente distinguibles: la de transición de estado a estado $\boldsymbol{t: Q \times \Gamma \rightarrow Q}$ y la del sentido del próximo movimiento del cabezal de entrada $\boldsymbol{m: Q \times \Gamma \rightarrow \{I, N, D\}}$.

e) El alfabeto de movimientos del cabezal contiene tres símbolos **{I, N, D}** que significan: izquierda (**I**), neutro (**N**) y derecha (**D**). Podría incluso agregarse un símbolo de detención de operaciones **P**; en el tratamiento de las máquinas de Turing en capítulos posteriores se efectuará este agregado y se comentará su función.

f) Un AFDB que contemple en todos los casos el movimiento del cabezal hacia la derecha se convierte en un AFD convencional, por lo que puede reconocerse a este último como un caso particular del primero.

Conceptos asociados a los AFDB

Aceptación de cadenas

Una cadena es aceptada por el AFDB cuando éste ha alcanzado un estado de aceptación y la cadena ha sido leída por lo menos una vez. Esto último significa que la cantidad de intervalos de tiempo demandados por la aceptación será igual o mayor al largo de la cadena en cinta. Nótese que si la cantidad de intervalos de tiempo necesarios para aceptar una cadena es igual a su largo, el AFDB se ha comportado como un AFD convencional.

Algunos autores como Cases Muñoz (2002) plantean la aceptación de una cadena cuando se ha alcanzado un estado de aceptación y el cabezal de lectura está a la derecha de la cadena de entrada (sobre el símbolo de fin de cinta, EOT), criterio que es compartido y adoptado en lo sucesivo en este texto.

Configuración

Al haber desaparecido en el AFDB el concepto de *cadena a ser leída*, debe replantearse la definición de configuración o descripción instantánea, que había sido propuesta para los AFD convencionales.

Luego, para la completa definición de la condición en la que se encuentra un AFDB en un instante dado se requieren tres componentes, que son: *i*) el estado actual, *ii*) el contenido de la cinta de entrada y *iii*) la posición del cabezal. Para esto último, se adopta la convención de que el símbolo de inicio de cinta (BOT) está en la posición **0** y el contador se incrementa hacia la derecha, lo que implica que el primer carácter de la cadena de entrada está en la posición **1**.

Según este criterio, la configuración K_t de un AFDB que opera sobre una cierta cadena de entrada α, que en el instante **t** está en el estado **q** y con el cabezal en la posición **k** es:

$$K_t = (q, \vdash\alpha\dashv, k)$$

Bajo esta definición, la configuración inicial toma la forma:

$$K_0 = (q_0, \vdash\alpha\dashv, 0)$$

y la configuración final de aceptación, bajo el criterio adoptado es:

$$K_A = (q_A, \vdash\alpha\dashv, n), \text{ en donde } q_A \in A \text{ y } n=|\alpha|+1$$

Una forma alternativa de representar la configuración de un AFDB es:

$$K_t = \delta q \beta$$

Donde **q** es el estado actual del autómata, δ el prefijo de la cadena de entrada α que antecede al cabezal y β el sufijo que sigue al cabezal $(\alpha=\delta\beta)$. Según este criterio, la configuración inicial es $K_0=q_0\alpha$ y la final $K_A=\alpha q_A$.

Lenguaje reconocido por el AFDB

Usando a la notación de movimientos entre configuraciones definida para los autómatas finitos, el lenguaje **L** que es aceptado por un autómata finito bidireccional **M** puede definirse en símbolos como:

$$L(M) = \{\alpha \ / \ \alpha \in \Sigma_E^* \text{ y } q_0\alpha \vdash^* \alpha q_A, \ q_A \in A\}$$

El problema de la parada

Ya fue anticipado que el AFDB puede quedar atrapado en un ciclo cerrado y esto puede ser reconocido, cuando al operar sobre una cadena de entrada, se repita una misma configuración. Siendo finitos el conjunto de estados posibles y la cinta de entrada, aunque con cierto esfuerzo, esta situación puede determinarse entonces por un algoritmo.

Extensión al tratamiento de palabras

La extensión de la función de transición para describir lo que ocurre a partir de cada estado si se recibe una secuencia concatenada de los símbolos de entrada en lugar de recibir símbolos aislados, es realizada al igual que en el AFD. Aunque no la explicitaremos aquí, es importante notar que en este caso la extensión debe incluir también la función de movimiento, ya que esta última es determinante en el comportamiento del AFDB.

Equivalencia entre AFDB y AFD

El movimiento del cabezal en dos sentidos no brinda ninguna capacidad adicional con respecto al AFD, lo que implica que todo AFDB tiene un AFD equivalente, lo que aquí no es demostrado. El razonamiento inverso ya fue establecido al admitirse que los AFD son un caso particular del AFDB.

Ejemplo 3.10

Proponer un AFDB que resuelva el problema de aceptar las cadenas que responden a la forma general $\alpha=(a+b)^*ba$ del Ejemplo 3.5.

En la solución que aquí se propone el autómata busca el final de la cadena y luego retrocede para asegurarse de que los dos últimos símbolos corresponden al sufijo **ba**. De esta manera, el problema se circunscribe a analizar este sufijo, prescindiendo totalmente del prefijo que lo antecede, que puede no existir. En caso de que el sufijo no cumpla la condición requerida el autómata arriba a un estado de error que aquí es identificado como c_4. El grafo del AFDB es el siguiente:

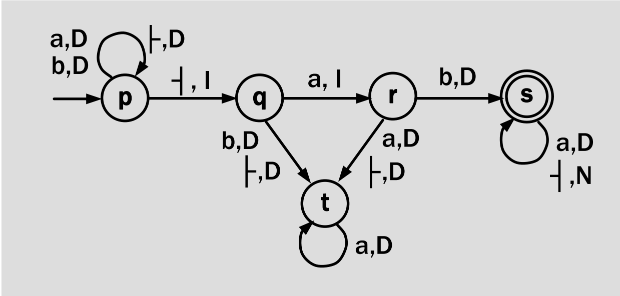

Figura 3.20: Grafo del AFDB del Ejemplo 3.10.

La definición formal del AFDB es la que se plantea a continuación:

AFDB=(Σ_E, Γ, Q, q_0, A, *f*)
Σ_E = {a, b}
Γ = {a, b, ├, ┤}
Q = {p, q, r, s, t}
q_0 = p
A = {s}

***f*:**	a	b	├	┤
→ p	p ,D	p ,D	p ,D	q ,I
q	r ,I	t ,D	t ,D	
r	t ,D	s ,D	t ,D	
* s	s ,D			s ,N
t	t ,D			

Tabla 3.21: Función de transición.

Nótese que la función ***f*** ha sido definida como parcial, obviando establecer transiciones innecesarias o para situaciones imposibles.

A efectos de estudiar el comportamiento del AFDB, se considera una cadena que responde a la forma general ya definida α, tal como **δ=abbba** y, para ello, se recurre al árbol de descripciones instantáneas o árbol de configuraciones, que es representado en la Figura 3.21.

En el árbol, puede verse que habiendo terminado de leer la cadena y estando el cabezal sobre el símbolo de fin de cinta, el autómata queda en un estado de aceptación, por lo cual la cadena es aceptada.

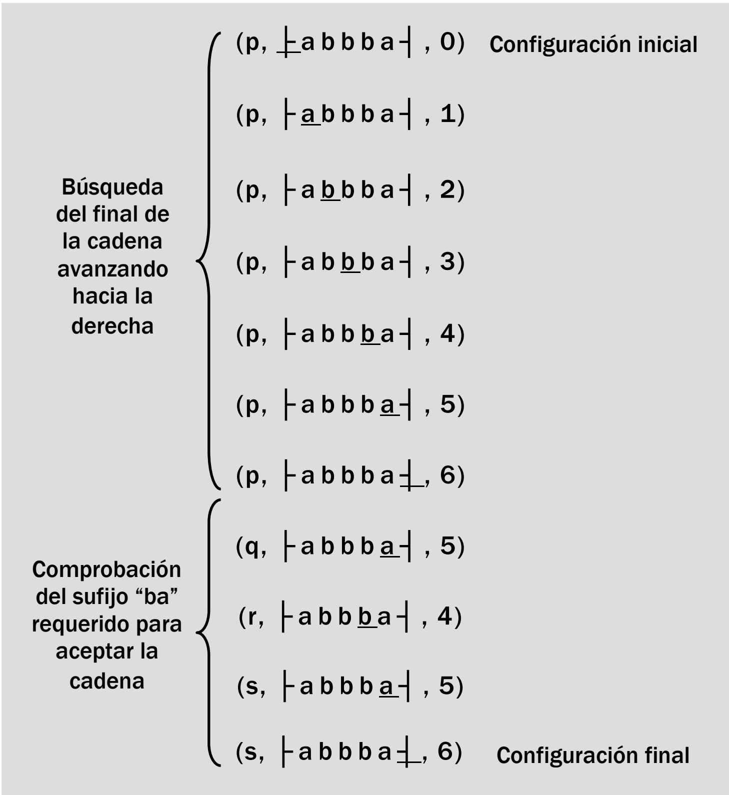

Figura 3.21: Árbol de configuraciones de aceptación de **abbba**.

Una alternativa muy conveniente para visualizar el comportamiento de un AFDB es utilizar el esquema que se representa en la Figura 3.22, donde la cinta es representada una única vez y se muestra el movimiento del cabezal sobre la misma. El símbolo correspondiente al estado de la máquina sirve a su vez para mostrar la posición del cabezal y los cambios de sentido se dibujan en diferentes líneas horizontales por razones de claridad. Ésta es una representación muy utilizada por la literatura para los autómatas que se mueven en dos sentidos.

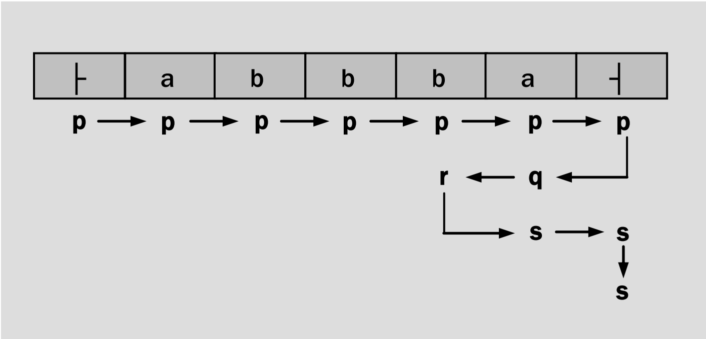

Figura 3.22: Esquema de movimientos al aceptarse la cadena **abbba**.

//