LÒGICA I LLENGUATGES

SOLUCIONS DE PROBLEMES

SETMANA DEL 10 D'ABRIL

Exercici 1 (llista 3). Sigui $\Sigma = \{0, 1, 2\}$. Siguin $L_1 = \{1, 02, 10\}$ i $L_2 = \{\lambda, 112, 0\}$. Determinar els llenguatges L_1L_2, L_2L_1, $L_1L_2 \cup L_2L_1$ i $L_1L_2 \cap L_2L_1$.

Solució : $L_1L_2 = \{1, 1112, 10, 02, 02112, 020, 10, 10112, 100\}$,

$L_2L_1 = \{1, 02, 10, 1121, 11202, 11210, 01, 002, 010\}$,

$L_1L_2 \cup L_2L_1 = \{1, 1112, 10, 02, 02112, 020, 10, 10112, 100, 1121, 11202, 11210, 01, 002, 010\}$,

$L_1L_2 \cap L_2L_1 = \{1, 10, 02\}$.

Exercici 2. Sobre l'alfabet $\Sigma = \{0, 1\}$, siguin:

$L_1 =$ el conjunt de paraules de bits que tenen exactament tants 0's com 1's,

$L_2 =$ el conjunt de paraules de bits que tenen almenys tants 0's com 1's,

$L_3 =$ el conjunt de paraules de bits que tenen almenys tants 1's com 0's.

Aleshores, determineu els llenguatges L_1L_1, L_1L_2, L_1L_3, L_2L_1, L_2L_2, L_2L_3, L_3L_1, L_3L_2, L_3L_3.

Solució : $L_1L_1 = L_1$, $L_1L_2 = L_2L_1 = L_2$, $L_1L_3 = L_3L_1 = L_3$, $L_2L_2 = L_2$, $L_3L_3 = L_3$, $L_2L_3 = \{0, 1\}^*$, $L_3L_2 = \{0, 1\}^*$.

Per a demostrar que $L_2L_3 = \{0, 1\}^*$, suposem que $x \in \{0, 1\}^*$. Aleshores, si $n_1(x) \leq n_0(x)$ tenim que $x = x \cdot \lambda \in L_2L_3$. I si $n_0(x) \leq n_1(x)$ tenim que $x = \lambda \cdot x \in L_2L_3$.

De manera anàloga, es demostra que $L_3L_2 = \{0, 1\}^*$.

Exercici 3 (llista 3). Considerem $L_1 = \{x \in \{0, 1\}^* : n_0(x)$ és parell $\}$ i $L_2 = \{x \in \{0, 1\}^* : n_0(x)$ és senar $\}$. Llavors, determineu L_1^* i L_2^*.

Solució : $L_1^* = L_1$, ja que si $x_1, \ldots, x_n$ són paraules tals que $n_0(x_i)$ és parell per $i = 1 \ldots n$, aleshores $n_0(x_1 x_2 \ldots x_n)$ és parell.

$L_2^* = \{x \in \{0,1\}^* : n_0(x) > 0\} \cup \{\lambda\}$, ja que si $x \in \{0,1\}^*$ tal que $n_0(x) > 0$, aleshores x és de la forma $1 \ldots 101 \ldots 10 \ldots 1 \ldots 101....1$ on els blocs de 1's poden ser buits. Aleshores, com que cada bloc $1 \ldots 10$ és una paraula de L_2 i el bloc final $1....101....1$ és també una paraula de L_2, la paraula x és una paraula de L_2^*.

Exercici 4 (llista 3). Demostreu que per a tot llenguatge L, $(L^*)^* = L^*$.

Solució : Clarament, $L^* \subseteq (L^*)^*$, ja que tot llenguatge està contingut en la seva clausura. Llavors, demostrem que $(L^*)^* \subseteq L^*$. Sigui $x \in (L^*)^*$. Aleshores, $x = x_1 \cdot \ldots \cdot x_n$ on $n \geq 0$ i $x_1, \ldots, x_n \in L^*$. Com tenim que cada $x_i \in L^*$, existeixen paraules $y_1^i, \ldots, y_{k_i}^i$ tals que $x_i = y_1^i \cdot \ldots y_{k_i}^i$. Per tant,

$$x = x_1 \cdot \ldots \cdot x_n = y_1^1 \cdot \ldots \cdot y_{k_1}^1 \cdot y_1^2 \cdot \ldots \cdot y_{k_2}^2 \cdot \ldots \cdot y_1^n \cdot \ldots \cdot y_{k_n}^n \in L^*.$$

Exercici 5 (llista 3). Determineu si són certes les següents conditions:
(a) $11001001 \in L((00 \cup 1)^*)$.
(b) $000 \in L((00 \cup 1)^*)$.
(c) $10100 \in L((00 \cup 1)^*)$.
(d) $10000111 \in L((00 \cup 1)^*)$.
(e) $L(1^*0^*) \cap L(0^*1^*) = L(0^* \cup 1^*)$.
(f) $L(0^*1^*) \cap L(2^*3^*) = \emptyset$.
(g) $0123 \in L((0(23)^*1)^*)$.

Solució : (a) és certa.

(b) és falsa, perquè els ceros han de sortir de dos en dos.

(c) és falsa, perquè els ceros han de sortir de dos en dos, i per tant no pot sortir un 0 entre dos uns.

(d) és certa.

(e) és certa.

(f) és falsa, perquè la paraula buida pertany al llenguatge $L(0^*1^*)$ i també al llenguatge $L(2^*3^*)$.

(g) és falsa, perquè si en una paraula del llenguatge apareix un símbol després d'un 1, aquest símbol ha de ser un 0.

Exercici 6 (llista 3). Determineu els llenguatges corresponents a les següents expressions regulars:
(a) 1^*10.
(b) $(0 \cup 1)^*1(0 \cup 1)^*$.
(c) $(0 \cup 10)(1 \cup 01)^*$.
(d) $(0^*10^*10^*1)0^*$.
(e) $(1 \cup 01)^*00(10 \cup 1)^*$.
(f) $0(0 \cup 1)^*0 \cup 1(0 \cup 1)^*1 \cup 0 \cup 1$.

Solució : (a) $\{1^n0 : n > 1\}$.

(b) $\{x \in \{0,1\}^* : x$ conté algun 1 $\}$.

(c) $\{xx_1 \dots x_n : n \geq 0, x = 0 \vee x = 10, x_i = 1 \vee x_i = 01$ per $i = 1, \dots, n\}$.

(d) $\{x \in \{0,1\}^* : x$ té exactament tres uns $\}$.

(e) $\{x \in \{0,1\}^* :$ en x apareix 00 exactament una vegada $\}$.

(f) $\{x \in \{0,1\}^* : x$ comença i acaba en el mateix símbol $\}$.

Exercici 7 (llista 3). Demostreu que els següents parells d'expressions regulars no són equivalents:
(a) $\alpha = (0 \cup 1)^*$, $\beta = (0)^* \cup (1)^*$.
(b) $\alpha = (0 \cup 1)^*$, $\beta = (01)^*$.
(c) $\alpha = 00^*1$, $\beta = 0^*1$.
(d) $\alpha = (0^*1)^*$, $\beta = (0 \cup 1)^*1$.

Solució : Per a demostrar que dos expressions regulars no són equivalents, hem de donar una paraula que estigui en el llenguatge d'una de les expressions regulars, però no estigui en el llenguatge de l'altra expressió regular. Llavors, tenim:

(a) $01 \in L(\alpha)$, però $01 \notin L(\beta)$, ja que les paraules de $L(\beta)$ són blocs de ceros o blocs d'uns.

(b) $1 \in L(\alpha)$, però $1 \notin L(\beta)$, ja que tota paraula de $L(\beta)$ té un nombre parell de símbols.

(c) $1 \in L(\beta)$, però $1 \notin L(\alpha)$ ja que tota paraula de $L(\alpha)$ comença en 0.

(d) $\lambda \in L(\alpha)$, però $\lambda \notin L(\beta)$, ja que tota paraula de $L(\beta)$ acaba en 1.

Exercici 8 (llista 3). Simplifica les següents expressions regulars, trobant per cadascuna d'elles una expressió regular més simple i equivalent.
(a) $(0 \cup \lambda)^*$.

(b) $(0 \cup \lambda)(0 \cup \lambda)^*$.
(c) $\lambda \cup 0^* \cup 1^* \cup (0 \cup 1)^*$.
(d) $0^*1 \cup (0^*1)0^*$.
(e) $(0^*1)^* \cup (1^*0)^*$.
(f) $(0 \cup 1)^*0(0 \cup 1)^*$.

Solució : Recordem que dos expressions regulars α i β són equivalents, si $L(\alpha) = L(\beta)$. Si dos expressions regulars α i β són equivalents, escrivim $\alpha \equiv \beta$. Llavors tenim:

(a) $(0 \cup \lambda)^* \equiv 0^*$.

(b) $(0 \cup \lambda)(0 \cup \lambda)^* \equiv 0^*$.

(c) $\lambda \cup 0^* \cup 1^* \cup (0 \cup 1)^* \equiv (0 \cup 1)^*$.

(d) $0^*1 \cup (0^*1)0^* \equiv 0^*10^*$.

(e) $(0^*1)^* \cup (1^*0)^* \equiv (0 \cup 1)^*$.

(f) $(0 \cup 1)^*0(0 \cup 1)^* \equiv (0 \cup 1)^*01^*$.

LÒGICA I LLENGUATGES

PROBLEMES

Llenguatges regulars

del 1 al 8 en el campus

Imprimir just amb les funcions

Exercici 1. Sigui $\Sigma = \{0, 1, 2\}$. Siguin $L_1 = \{1, 02, 10\}$ i $L_2 = \{\lambda, 112, 0\}$. Determinar els llenguatges L_1L_2, L_2L_1, $L_1L_2 \cup L_2L_1$ i $L_1L_2 \cap L_2L_1$.

Exercici 2. Sobre l'alfabet $\Sigma = \{0, 1\}$, siguin:

$L_1 =$ el conjunt de paraules de bits que tenen exactament tants 0's com 1's,

$L_2 =$ el conjunt de paraules de bits que tenen almenys tants 0's com 1's,

$L_3 =$ el conjunt de paraules de bits que tenen almenys tants 1's com 0's.

Aleshores, determineu els següents L_1L_1, L_1L_2, L_1L_3, L_2L_1, L_2L_2, L_2L_3, L_3L_1, L_3L_2, L_3L_3.

Exercici 3. Considerem els llenguatges $L_1 = \{x \in \{0, 1\}^* : n_0(x) \text{ és parell }\}$ i $L_2 = \{x \in \{0, 1\}^* : n_0(x) \text{ és senar }\}$. Determineu L_1^* i L_2^*.

Exercici 4. Demostreu que per a tot llenguatge L, $(L^*)^* = L^*$.

Exercici 5. Determineu si son certes les següents condicions:
(a) $11001001 \in L((00 \cup 1)^*)$.
(b) $000 \in L((00 \cup 1)^*)$.
(c) $1101100 \in L((00 \cup 1)^*)$.
(d) $10000111 \in L((00 \cup 1)^*)$.
(e) $L(1^*0^*) \cap L(0^*1^*) = L(0^* \cup 1^*)$.
(f) $L(0^*1^*) \cap L(2^*3^*) = \emptyset$.
(g) $0123 \in L((0(23)^*1)^*)$.

Exercici 6 Determineu els llenguatges corresponents a les següents expressions regulars:
(a) 1^*10.

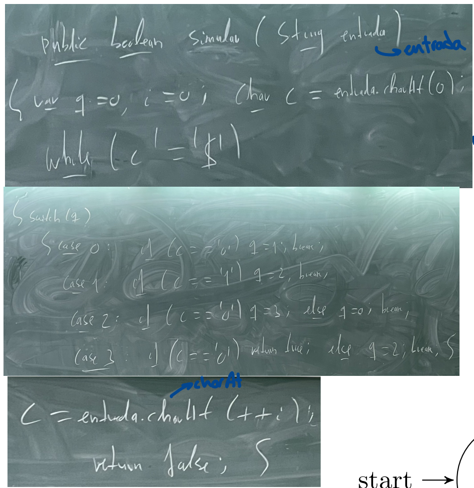
(b) $(0 \cup 1)^*1(0 \cup 1)^*$.
(c) $(0 \cup 10)(1 \cup 01)^*$.
(d) $(0^*10^*10^*1)0^*$.
(e) $(1 \cup 01)^*00(10 \cup 1)^*$.
(f) $0(0 \cup 1)^*0 \cup 1(0 \cup 1)^*1 \cup 0 \cup 1$.

Exercici 7. Demostrar que els següents parells de'expressions regulars no son equivalents:
(a) $\alpha = (0 \cup 1)^*$ y $\beta = 0^* \cup 1^*$.
(b) $\alpha = (0 \cup 1)^*$ y $\beta = (01)^*$.
(c) $\alpha = 00^*1$ y $\beta = 0^*1$.
(d) $\alpha = (0^*1)^*$ y $\beta = (0 \cup 1)^*1$.

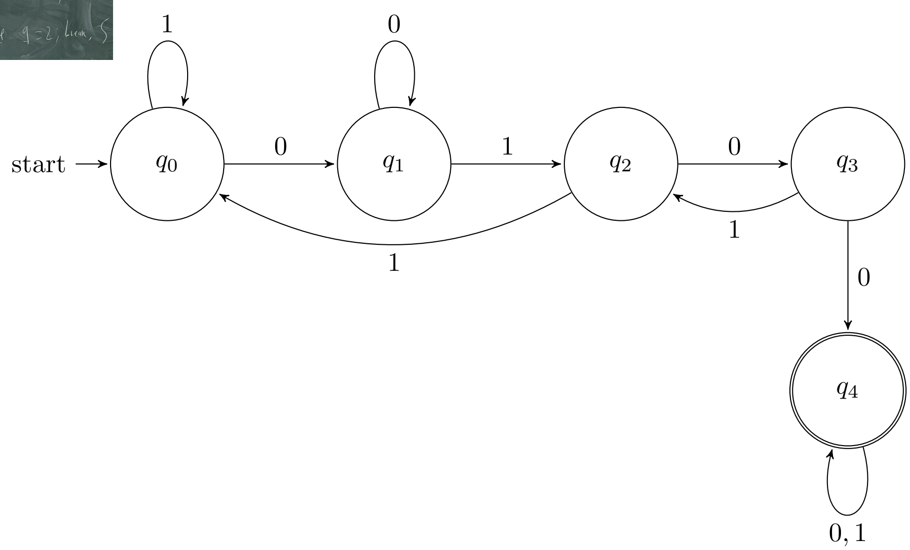
Exercici 8. Simplifica les següents expressions regulars, trobant per cadascuna d'elles una expressió regular més simple i equivalent.
(a) $(0 \cup \lambda)^*$.
(b) $(0 \cup \lambda)(0 \cup \lambda)^*$.
(c) $\lambda \cup 0^* \cup 1^* \cup (0 \cup 1)^*$.
(d) $0^*1 \cup (0^*1)0^*$.
(e) $(0^*1)^* \cup (1^*0)^*$.
(f) $(0 \cup 1)^*0(0 \cup 1)^*$.

Exercici 9. Considerem el següent autòmat determinista M, on q_4 es l'únic estat acceptador:

(c)



'$' como final de la palabra



Llavors, es demana:

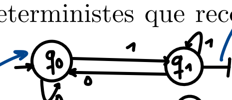
(a) Descriure $L(M)$ informalment. $L(M) = \{x \in \{0,1\}^* : x$ contiene 0100 como subpalabra$\}$

(b) Descriure $L(M)$ mitjançant una expressió regular. $L(M) = L(\varphi)$ donde $\varphi = (0 \vee 1)^* 0100 (0 \vee 1)^*$

(c) Simular M mitjançant un programa en JAVA.

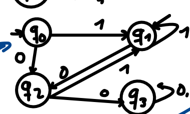
Exercici 10. Construir autòmats deterministes que reconeguin els següents llenguatges:

(a) $\{x \in \{0,1\}^* : x$ acaba en $1\}$.

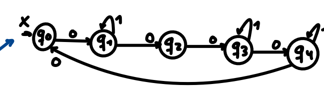


q_0: último bit leído es 0
q_1: " " " " 1

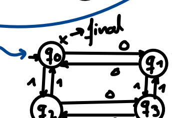
(b) $\{x \in \{0,1\}^* : x$ acaba en 1 i no conté 00$\}$.



(c) $\{x \in \{0,1\}^* : n_0(x)$ és un múltiple de 5$\}$.



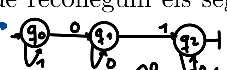
(d) $\{x \in \{0,1\}^* : n_0(x)$ és parell i $n_1(x)$ és parell $\}$.



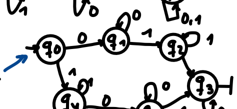
q_0: $n_0(x)$ es par, $n_1(x)$ es par
q_1: $n_0(x)$ es impar, $n_1(x)$ es par
q_2: $n_1(x)$ es impar, $n_0(x)$ es par
q_3: $n_0(x)$ es impar, $n_1(x)$ es impar

Exercici 11. Construir autòmats deterministes que reconeguin els següents llenguatges:

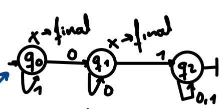
(a) $\{x \in \{0,1\}^* : x$ conté com a subparaula 01$\}$.



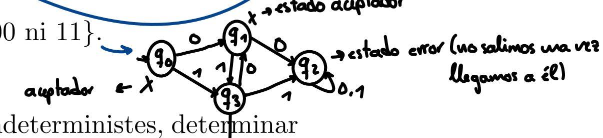
(b) $\{x \in \{0,1\}^* : x$ conté com subparaules 01 i 10$\}$.



(c) $\{x \in \{0,1\}^* : x$ no conté com a subparaula 01$\}$.



(d) $\{x \in \{0,1\}^* : x$ no conté com subparaules ni 00 ni 11$\}$.

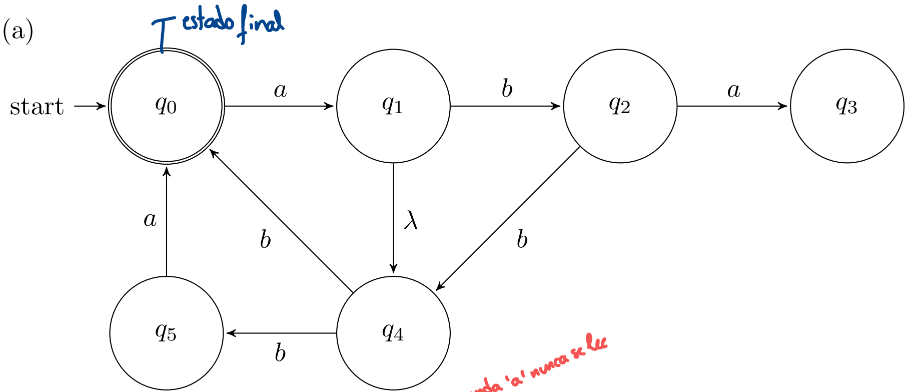


Exercici 12. Per cada un dels següents autòmats indeterministes, determinar (*)

(*) Un autómata indeterminista es una estructura $M = (K, I, \Delta, q_0, F)$ donde K, I, q_0, F son como en la definición de AD y Δ es un subconjunto de $K \times (\Sigma \cup \{\lambda\}) \times K$. (→delta)
En el caso determinista, Δ es una función de $K \times \Sigma$ en K.

les paraules de les llistes corresponents que són reconegudas:

(a)



$q_0 \to q_1 \to q_4 \to q_0$ ×2 la segunda 'a' nunca se lee

$q_0 \to q_1 \to q_4 \to q_5 \to q_0$ ✓ (2n recorrido 3 veces)

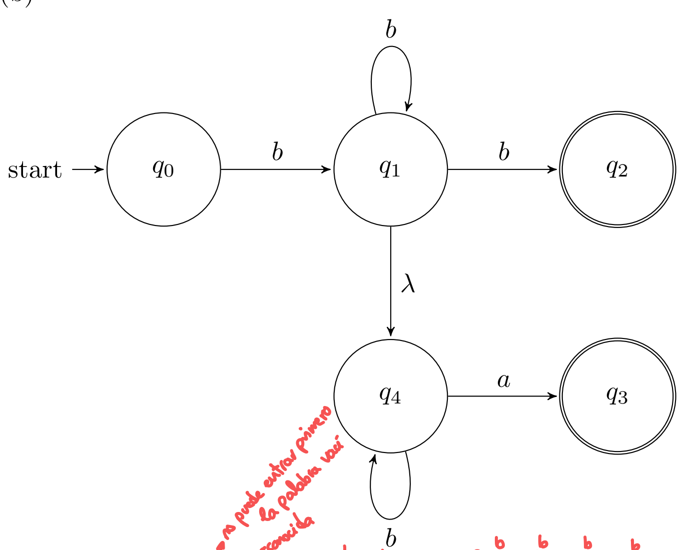
✓ ✓ ✓ ✗ ✓ ✗ ✓ ✓

Llista de paraules: λ, ab, $abab$, aaa, aba, abb, ab, $ababab$.

la segunda 'b' no se lee nunca

palabra vacía (transición sin leer símbolo de entrada)

(b)



no puede entrar primero la palabra vacía

no reconocida

$q_0 \overset{b}{\mapsto} q_1 \overset{b}{\mapsto} q_2$

$q_0 \overset{b}{\mapsto} q_1 \overset{b}{\mapsto} q_1 \overset{b}{\mapsto} q_1 \overset{b}{\mapsto} q_2$

✗ ✗ ✓ ✓ ✓ ✗

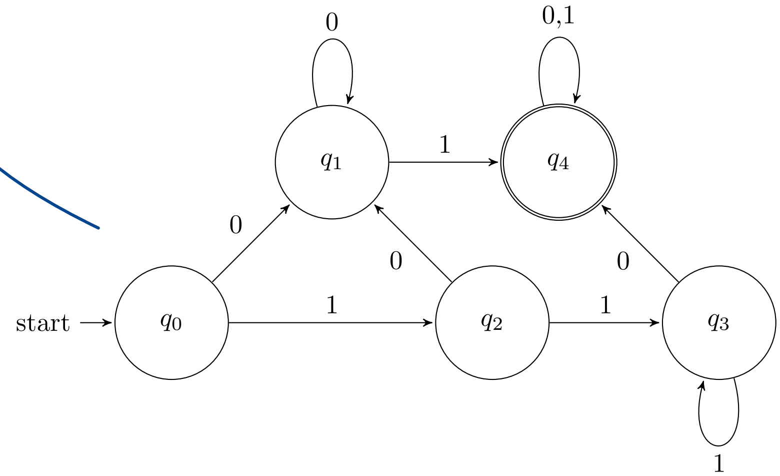
Llista de paraules: λ, b, bb, $bbba$, $bbbb$, $baaa$.

se bloquea en q_1 o q_4

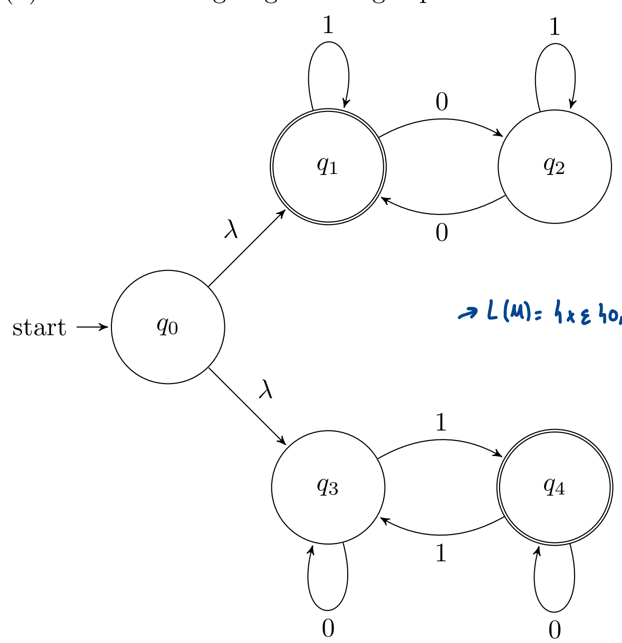
$q_0 \overset{b}{\mapsto} q_1 \overset{\lambda}{\mapsto} q_4 \overset{b}{\mapsto} q_4 \overset{a}{\mapsto} q_2$

<u>Exercici 13</u>. (a) Descriu el llenguatge reconegut per l'autòmat determinista següent:

L(M) = {x̄ ∈ {0,1}*: x contiene '01' o '110'}



(b) Descriu el llenguatge reconegut per l'autòmat indeterminista següent:



→ L(M) = {x ∈ {0,1}*: $m_0(x)$ es par, $m_1(x)$ es impar.

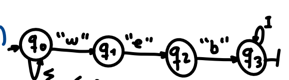
Σ = conjunto de caràcteres

Exercici 14. (a) Definir un autòmat indeterminista que determini si en un text apareix la paraula "web" .

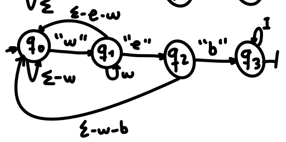
(b) Convertir directament l'autòmat de l'apartat (a) en un autòmat determinista.

(c) Simular l'autòmat del apartat (b) mitjançant un programa en JAVA.

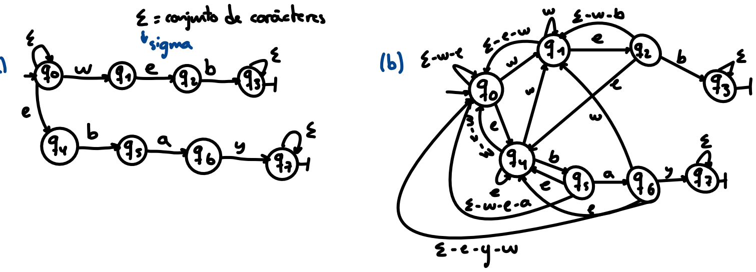
(a)



(b)

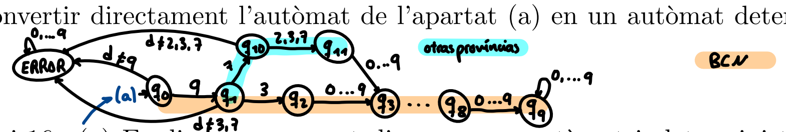


(c)



Exercici 15. (a) Definir un autòmat indeterminista per determinar si en un text apareix la paraula "web" o la paraula "ebay".

(b) Convertir directament l'autòmat de l'apartat (a) en un autòmat determinista.



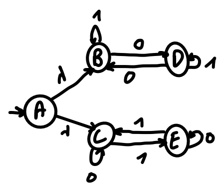
Exercici 16. (a) Explicar com es pot dissenyar un autòmat indeterminista per reconèixer els números de telèfon de les províncies de Catalunya.

(b) Explicar com a partir de l'autòmat de l'apartat (a), es pot dissenyar un programa en JAVA per reconèixer aquests números.

Exercici 17. Mitjançant l'algorisme vist a classe, construir un autòmat determinista equivalent a l'autòmat indeterminista $M = (\{A, B, C, D, E\}, \{0, 1\}, \Delta, A, \{B, C\})$ on Δ està definida per la següent taula:

$A \xrightarrow{0} \emptyset$ | $A \xrightarrow{1} \emptyset$
$B \xrightarrow{0} D$ | $B \xrightarrow{1} B$
$C \xrightarrow{0} C$ | $C \xrightarrow{1} E$
$D \xrightarrow{0} B$ | $D \xrightarrow{1} D$
$E \xrightarrow{0} E$ | $E \xrightarrow{1} C$

A	λ	B
A	λ	C
B	0	D
B	1	B
C	0	C
C	1	E
D	0	B
D	1	D
E	0	E
E	1	C



$\Lambda(A) = ABC =: q_0$ $K' = \{ABC, BC, BE, DC, DE\}$
$\Lambda(B) = B$ $q'_0 = ABC$ $F' = \{ABC, BC, BE, DC\}$
$\Lambda(C) = C$
$\Lambda(D) = D$
$\Lambda(E) = E$

$\delta'(ABC, 0) = \Lambda(\{C\} \cup \{D\}) = DC$
$\delta'(ABC, 1) = \Lambda(\{B\} \cup \{E\}) = BE$
$\delta'(DC, 0) = \Lambda(\{B\} \cup \{C\}) = BC$
$\delta'(DC, 1) = \Lambda(\{D\} \cup \{E\}) = DE$
$\delta'(BE, 0) = \Lambda(\{D\} \cup \{E\}) = DE$
$\delta'(BE, 1) = \Lambda(\{B\} \cup \{C\}) = BC$
$\delta'(DE, 0) = \Lambda(\{B\} \cup \{E\}) = BE$
$\delta'(DE, 1) = \Lambda(\{D\} \cup \{C\}) = DC$
$\delta'(BC, 0) = \Lambda(\{D\} \cup \{C\}) = DC$
$\delta'(BC, 1) = \Lambda(\{B\} \cup \{E\}) = BE$

Exercici 18. Mitjançant l'algorisme vist a classe, construir un autòmat determinista equivalent a l'autòmat indeterminista $M = (\{A, B, C, D\}, \{0, 1\}, \Delta, A, \{A, D\})$ on Δ està definida per la següent taula:

$\Lambda(A) = ACD$,
$\Lambda(B) = B$,
$\Lambda(C) = CD$,
$\Lambda(D) = D$

$A \xrightarrow{0} \emptyset$ | $A \xrightarrow{1} B$
$B \xrightarrow{0} B, C$ | $B \xrightarrow{1} C$
$C \xrightarrow{0} A$ | $C \xrightarrow{1} \emptyset$
$D \xrightarrow{0} B$ | $D \xrightarrow{1} B$

A	1	B
A	λ	C
B	0	B
B	0	C
B	1	C
C	0	A
C	λ	D
D	0	B
D	1	B

Estado inicial = $\Lambda(A) = ACD$

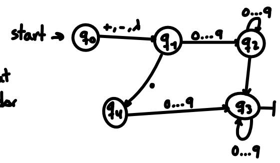
$\delta'(ACD, 0) = \Lambda(A) \cup \Lambda(B) = ABCD$,
$\delta'(ACD, 1) = \Lambda(B) = B$,
$\delta'(ABCD, 0) = \Lambda(A) \cup \Lambda(B) \cup \Lambda(C) = ABCD$,
$\delta'(ABCD, 1) = \Lambda(B) \cup \Lambda(C) = BCD$,
$\delta'(B, 0) = \Lambda(B) \cup \Lambda(C) = BCD$,
$\delta'(B, 1) = \Lambda(C) = CD$

$\delta'(BCD, 0) = \Lambda(A) \cup \Lambda(B) \cup \Lambda(C) = ABCD$,
$\delta'(BCD, 1) = \Lambda(B) \cup \Lambda(C) = BCD$,
$\delta'(CD, 0) = \Lambda(A) \cup \Lambda(B) = ABCD$,
$\delta'(CD, 1) = \Lambda(B) = B$.

Estados aceptadores: ACD, ABCD, BCD, CD
Estados: B, ACD, ABCD, BCD, CD.

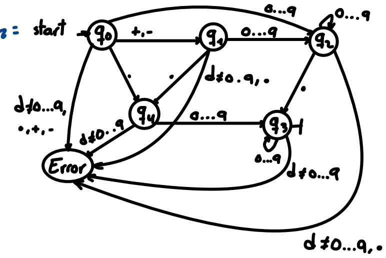
Exercici 19. Mitjançant l'algorisme vist a classe, construir un autòmat

a) M_1 =



q_3 = únic estat acceptador

b) M_2 =



determinista equivalent a l'autòmat indeterminista $M = (\{q_0, q_1, , q_2, q_3, q_4\}, \{0, 1\}, \Delta, q_0, \{q_4\})$ on Δ està definida per la següent taula:

$\Lambda(q_i) = q_i$ para $i = 1,2,3,4$

$q_0 \overset{0}{\mapsto} q_4$	$q_0 \overset{1}{\mapsto} q_1, q_2$
$q_1 \overset{0}{\mapsto} \emptyset$	$q_1 \overset{1}{\mapsto} q_0$
$q_2 \overset{0}{\mapsto} \emptyset$	$q_2 \overset{1}{\mapsto} q_3$
$q_3 \overset{0}{\mapsto} q_0$	$q_3 \overset{1}{\mapsto} \emptyset$
$q_4 \overset{0}{\mapsto} \emptyset$	$q_4 \overset{1}{\mapsto} \emptyset$

q_0	1	q_1
q_0	1	q_2
q_0	0	q_4 → único estado aceptador
q_1	1	q_0
q_2	1	q_3
q_3	0	q_0

Estado inicial = $\Lambda(q_0) = q_0$

$\delta'(q_0, 0) = q_4$, $\delta'(q_0, 1) = q_1 q_2$, $\delta'(q_4, 0) = \emptyset$, $\delta'(q_4, 1) = \emptyset$, $\delta'(q_1 q_2, 0) = \emptyset$, $\delta'(q_1 q_2, 1) = \Lambda(q_0) \cup \Lambda(q_3) = q_0 q_3$,

$\delta'(q_0 q_3, 0) = q_0 q_4$, $\delta'(q_0 q_3, 1) = q_1 q_2$, $\delta'(q_0 q_4, 0) = q_4$, $\delta'(q_0 q_4, 1) = q_1 q_2$, $\delta'(\emptyset, 0) = \delta'(\emptyset, 1) = \emptyset$.

Estados: $q_0, q_4, q_1 q_2, \emptyset, q_0 q_4, q_0 q_3$.

Estados aceptadores: $q_4, q_0 q_4$.

<u>Exercici 20</u>. (a) Construir un autòmat indeterminista per reconèixer nombres decimals que continguin: (a) un signe + o − opcional; (b) una paraula de dígits; (c) un punt decimal; (d) una segona paraula de dígits. Tant la primera paraula de dígits com la segona poden estar buides, però almenys una de les dues paraules no pot estar buida.

(b) Explicar com es pot dissenyar un programa en JAVA per reconèixer nombres decimals.

<u>Exercici 21</u>. Modificar l'autòmat vist a classe per dissenyar l'analitzador lèxic d'un compilador, de manera que es reconeguin també els nombres decimals segons la definició donada en l'exercici 20.

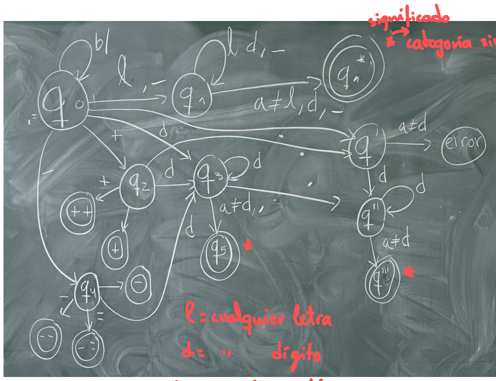
<u>Exercici 22</u>. Explicar com dissenyar un analitzador lèxic per reconèixer les següents categories sintàctiques:

(1) identificadors formats per lletres i dígits de manera que el primer caràcter és una lletra,

(2) nombres enters,

(3) els operadors aritmètics + i −,

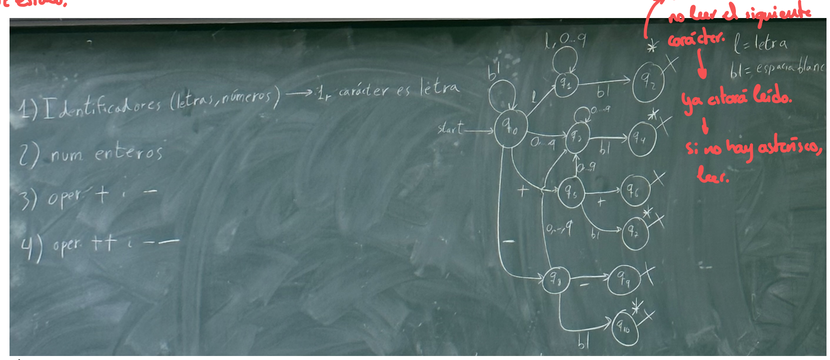
(4) els operadors ++ i −−.



significado

* categoría sintáctica que lee el carácter siguiente. Al programarlo y llegar a este estado, no leeríamos el siguiente.

bl : carácter en blanco

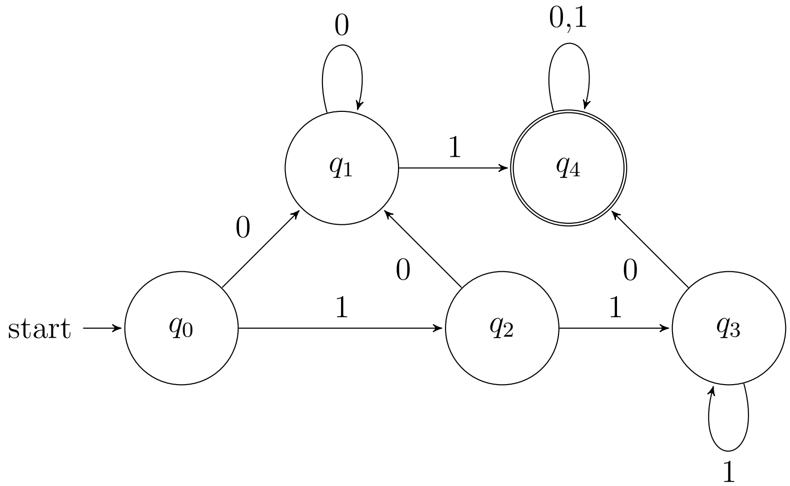


estado aceptador y no leer el siguiente carácter.

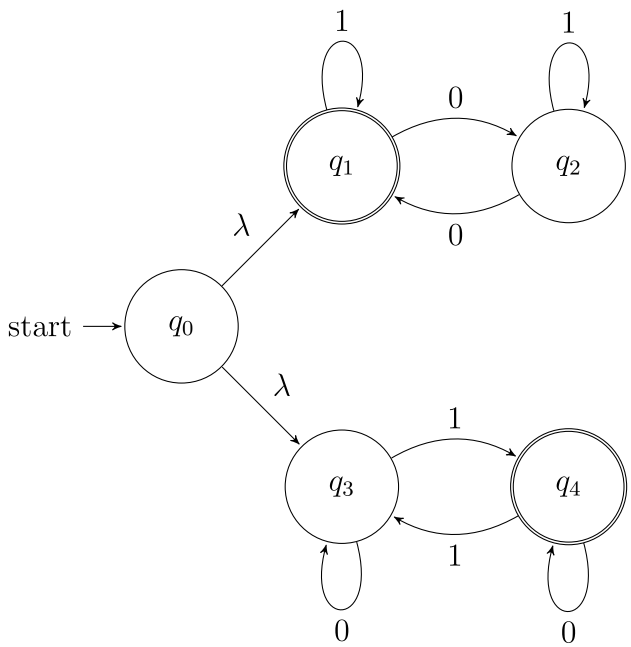
ya estará leído.

si no hay asterisco, leer.

Problema 1 . (a) Describir el lenguaje reconocido por el siguiente autómata determinista M, donde q_4 es el único estado aceptador.



(b) Describir el lenguaje reconocido por el siguiente autómata indeterminista, donde q_1 y q_4 son los estados aceptadores.



Solución:

(a) $L(M) = \{x \in \{0,1\}^* : x$ contiene 01 o 110 como subpalabra$\}$. Tambiíen, $L(M)$ se puede representar por la expresión regular $\alpha = (0 \cup 1)^*(01 \cup 110)(0 \cup 1)^*$.

(b) El estado q_1 reconoce el lenguaje de las palabras de bits en las cuales el número de ceros es par. Y el estado q_4 reconoce el lenguaje de las palabras de bits en las cuales el número de unos es impar. Por tanto, como q_1 y q_4 son los estados aceptadores, se tiene que el lenguaje reconocido por el autómata es $L = \{x \in \{0,1\}^* : n_0(x)$ es par ó $n_1(x)$ es impar $\}$.

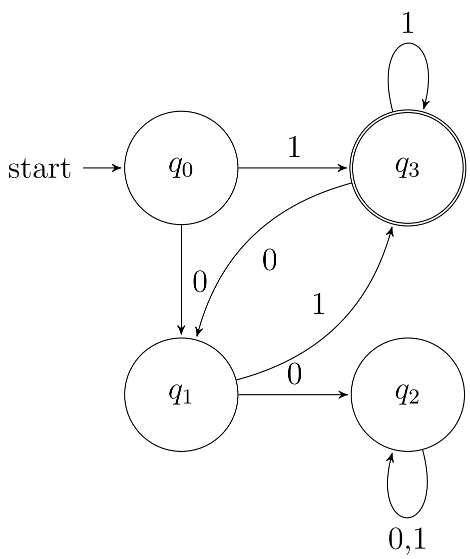
Problema 2. Definir autómatas deterministas que reconozcan los siguientes lenguajes.

(a) $\{x \in \{0,1\}^* : x$ acaba en 1 y no contiene $00\}$.

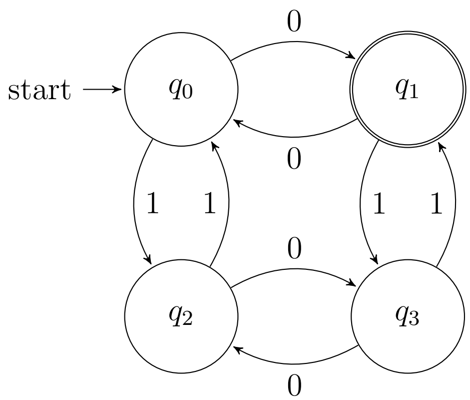
(b) $\{x \in \{0,1\}^* : x$ tiene un número impar de ceros y un número par de unos $\}$.

Solución:

(a) Definimos el siguiente autómata determinista, que consta de los estados q_0, q_1, q_2 y q_3, donde q_0 es el estado inicial y q_3 es el único estado aceptador.



(b) Definimos el siguiente autómata determinista, que consta de los estados q_0, q_1, q_2 y q_3, donde q_0 es el estado inicial y q_1 es el único estado aceptador.



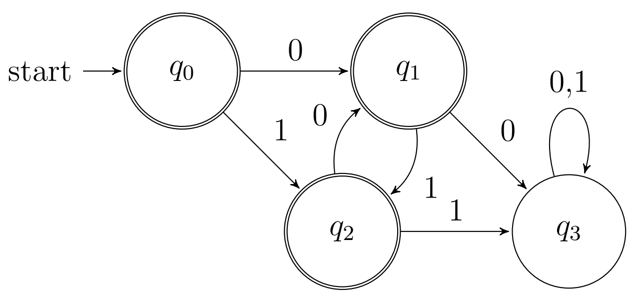
Obsérvese que estaremos en el estado q_0, cuando el número de ceros leídos sea par y el número de unos leídos sea par; estaremos en el estado q_1, cuando el número de ceros leídos sea impar y el número de unos leídos sea par; estaremos en el estado q_2, cuando el número de ceros leídos sea par y el número de unos leídos sea impar; y estaremos en el estado q_3, cuando el número de ceros leídos sea impar y el número de unos leídos sea impar. Como q_1 es el único estado aceptador del autómata, se tiene que el lenguaje del autómata es el conjunto $\{x \in \{0, 1\}^* : x$ tiene un número impar de ceros y un número par de unos $\}$.

Problema 3 . (a) Definir un autómata determinista M tal que $L)(M) = \{x \in \{0,1\}^* : x \text{ no contiene ni 00 ni 11}\}$.

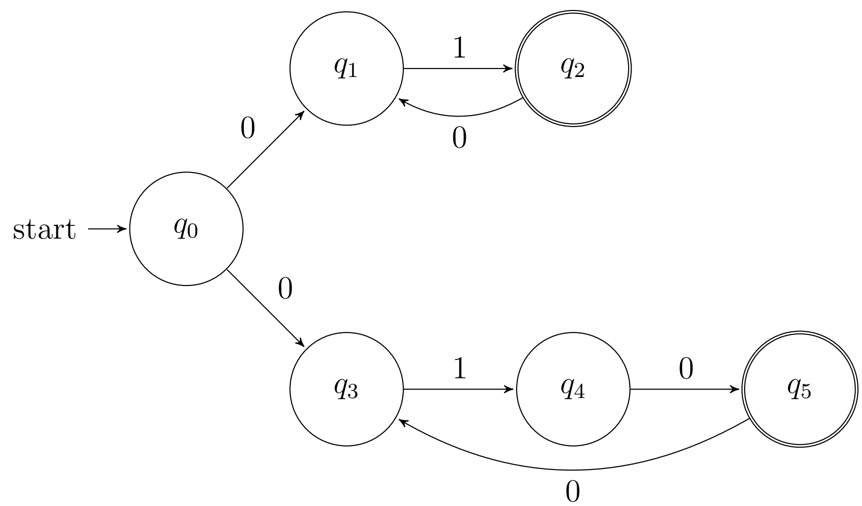
(b) Definir un autómata indeterminista M tal que $L(M) = \{(01)^n : n \geq 1\} \cup \{(010)^n : n \geq 1\}$, es decir, $L(M)$ es el lenguaje de las palabras formadas por 01 repetido una o más veces, o por 010 repetido una o más veces.

Solución:

(a) Definimos el siguiente autómata determinista, que consta de los estados q_0, q_1, q_2 y q_3, donde q_0 es el estado inicial y q_0, q_1 y q_2 son los estados aceptadores.



(b) Definimos el siguiente autómata indeterminista, que consta de los estados q_0, q_1, q_2, q_3, q_4 y q_5, donde q_0 es el estado inicial y q_2 y q_5 son los estados aceptadores.


start
q_0
0
q_1
1
q_2
0
0
q_3
1
q_4
0
q_5
0

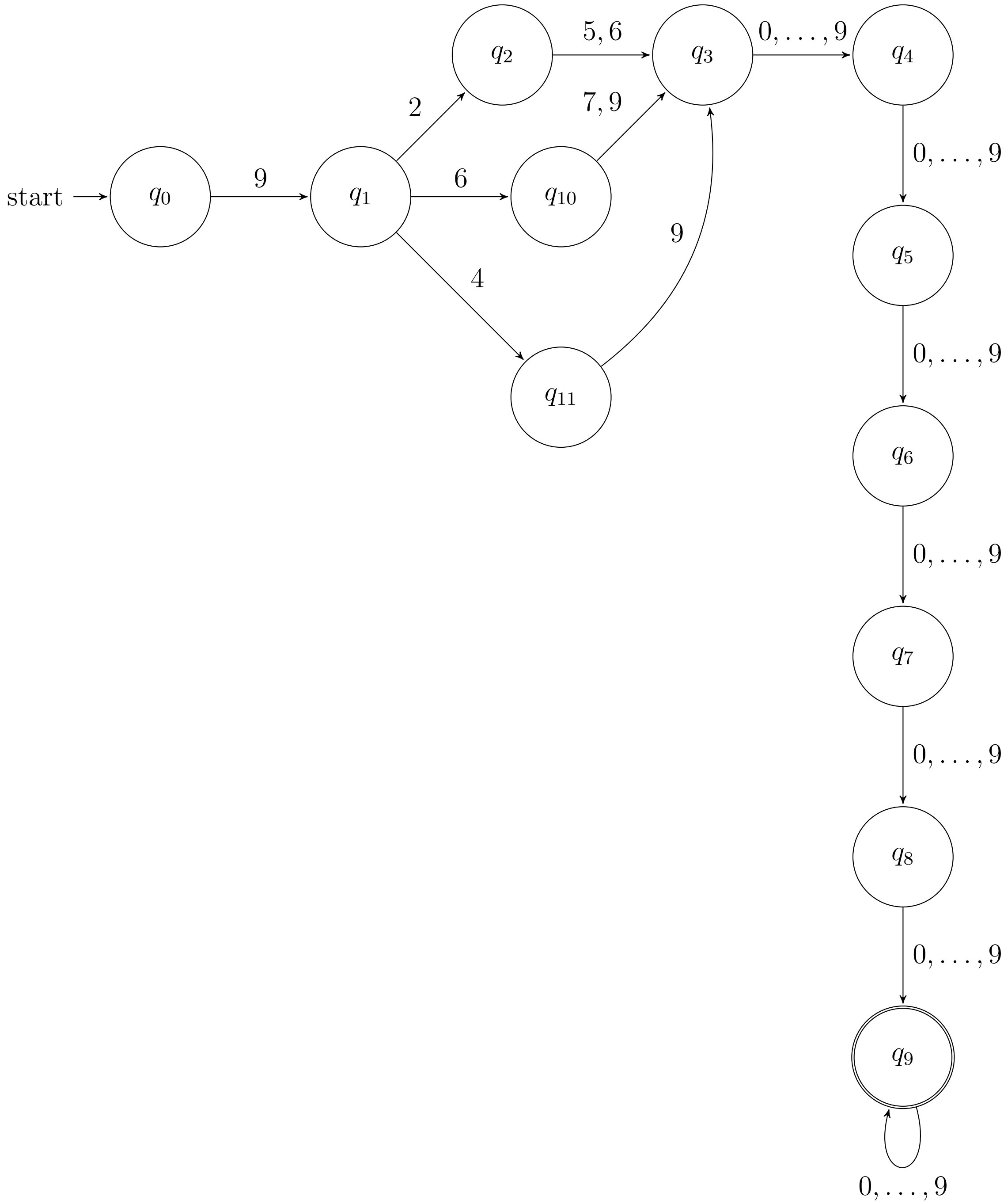
Problema 4. Para llamar por teléfono a una provincia de la comunidad de Castilla-La Mancha hay que marcar un número de seis dígitos y previamente el siguiente prefijo: 967 si se llama a Albacete, 969 si se llama a Cuenca, 949 si se llama a Guadalajara, 925 si se llama a Toledo, y 926 si se llama a Ciudad Real. Se pide entonces:

(a) Definir un autómata indeterminista que reconozca los números de teléfono de la comunidad de Castilla-La Mancha.

(b) Explicar cómo, utilizando el autómata del apartado (a), se puede escribir un programa en JAVA para reconocer los números de teléfono de Castilla-La Mancha.

Solución:

(a) El siguiente autómata indeterminista, en donde q_9 es el único estado aceptador, reconoce los números de teléfono de Castilla-La Mancha.


start
q_0
9
q_1
2
6
4
q_2
q_{10}
q_{11}
5, 6
7, 9
9
q_3
0, . . . , 9
q_4
0, . . . , 9
q_5
0, . . . , 9
q_6
0, . . . , 9
q_7
0, . . . , 9
q_8
0, . . . , 9
q_9
0, . . . , 9

(b) Para escribir un programa en JAVA que reconozca los números de teléfono de Castilla-La Mancha, en primer lugar hemos de eliminar el indeterminismo del autómata del apartado (a). Podemos hacer esto directamente añadiendo un estado de error al autómata, al cual llegaremos desde cualquier otro estado cuando entre un dígito distinto a los indicados en las transiciones del gráfico del autómata del apartado (a). El programa asociado a este autómata determinista será entonces el programa buscado para reconocer los números de teléfono de Castilla-La Mancha.

Problema 5. Consideremos el autómata indeterminista $M = (\{P, Q, R, S\}, \{a, b\}, \Delta, P, \{P, Q\})$ donde Δ está definida por la siguiente tabla:

P	a	S
P	a	Q
P	λ	Q
Q	b	Q
Q	λ	R
R	b	P
S	a	S
S	b	R

Siguiendo el método visto en clase, trasformar el autómata M en un autómata determinista equivalente.

Solución:

Calculamos primero los cierres de los estados. Se tiene que $\Lambda(P) = PQR$, $\Lambda(Q) = QR$, $\Lambda(R) = R$ y $\Lambda(S) = S$. Construimos entonces el autómata determinista M' equivalente a M. El estado inicial de M' es $\Lambda(P) = PQR$. Construimos la función de transición δ' para M'.

$\delta'(PQR, a) = \Lambda(Q) \cup \Lambda(S) = QRS,$
$\delta'(PQR, b) = \Lambda(P) \cup \Lambda(Q) = PQR,$
$\delta'(QRS, a) = \Lambda(S) = S,$
$\delta'(QRS, b) = \Lambda(P) \cup \Lambda(Q) \cup \Lambda(R) = PQR,$
$\delta'(S, a) = \Lambda(S) = S,$
$\delta'(S, b) = \Lambda(R) = R,$
$\delta'(R, a) = \emptyset,$
$\delta'(R, b) = \Lambda(P) = PQR,$
$\delta'(\emptyset, 0) = \delta'(\emptyset, 1) = \emptyset.$

Por tanto, los estados de M' son: PQR, QRS, S, R y $\emptyset$. Como P y Q son los estados aceptadores de M, los estados aceptadores de M' son PQR y QRS.

Problema 6. Consideremos el autómata indeterminista $M = (\{A, B, C, D\}, \{0, 1\}, \Delta, A, \{D\})$ donde Δ está definida por la siguiente tabla:

A	0	A
A	λ	B
B	0	C
B	λ	D
C	1	B
D	0	D

Se pide entonces:

(1) Describir el lenguaje $L(M)$.

(2) Siguiendo el método visto en clase, transformar el autómata M en un autómata determinista equivalente.

(3) Programar en JAVA o en C el autómata determinista obtenido en (2).

Solución:

(1) $L(M) = \{0^m(01)^n0^k : m, n, k \geq 0\}$. Tambiíen, $L(M)$ se puede representar por la expresión regular $0^*(01)^*0^*$.

(2) Se tiene que $\Lambda(A) = ABD$, $\Lambda(B) = BD$, $\Lambda(C) = C$ y $\Lambda(D) = D$. Construimos entonces el autómata determinista M' equivalente a M. El estado inicial de M' es $\Lambda(A) = ABD$. Construimos la función de transición δ' para M'.

$\delta'(ABD, 0) = \Lambda(A) \cup \Lambda(C) \cup \Lambda(D) = ABCD$,
$\delta'(ABD, 1) = \emptyset$,
$\delta'(ABCD, 0) = \Lambda(A) \cup \Lambda(C) \cup \Lambda(D) = ABCD$,
$\delta'(ABCD, 1) = \Lambda(B) = BD$,
$\delta'(\emptyset, 0) = \delta'(\emptyset, 1) = \emptyset$,
$\delta'(BD, 0) = \Lambda(C) \cup \Lambda(D) = CD$,
$\delta'(BD, 1) = \emptyset$,
$\delta'(CD, 0) = \Lambda(D) = D$,
$\delta'(CD, 1) = \Lambda(B) = BD$,
$\delta'(D, 0) = \Lambda(D) = D$,

$\delta'(D, 1) = \emptyset$.

Por tanto, los estados de M' son: $ABD, ABCD, BD, CD, D$ y $\emptyset$. Como D es el único estado aceptador de M, los estados aceptadores de M' son $ABD, ABCD, BD, CD$ y D (es decir, todos los estados salvo $\emptyset$).

(3) Representamos al estado ABD por 0, al estado $ABCD$ por 1, al estado BD por 2, al estado CD por 3 y al estado D por 4. Como $\emptyset$ es un estado de error, no hace falta representarlo. Podemos escribir entonces el siguiente programa en JAVA para simular el autómata M':

```
public boolean simular (String entrada)
{ int q = 0, i = 0;
  char c = entrada.charAt(0);
  while (c != '$')
  { switch(q)
        { case 0:
        if (c == '0') q = 1; else return false;
        break;
        case 1:
        if (c == '1') q = 2;
        break;
        case 2:
        if (c == '0') q = 3; else return false;
        break;
        case 3:
        if (c == '0') q = 4; else q = 2;
        break;
        case 4:
        if (c == '1') return false;
        break;}
  c = entrada.charAt(++i); }
  return true; }
```

Problema 7. Consideremos el autómata indeterminista $M = (\{A, B, C, D, E\}, \{0,1\}, \Delta, A, \{B, C\})$ donde Δ está definida por la siguiente tabla:

A	λ	B
A	λ	C
B	0	D
B	1	B
C	0	C
C	1	E
D	0	B
D	1	D
E	0	E
E	1	C

Se pide entonces:

(1) Describir el lenguaje $L(M)$.

(2) Siguiendo el método visto en clase, transformar el autómata M en un autómata determinista equivalente.

(3) Programar en JAVA el autómata determinista obtenido en (2).

Solución:

(1) Se observa que el estado B reconoce las palabras de bits que tienen un número par de ceros, y el estado C reconoce las palabras de bits que tienen un número par de unos. Por tanto, como B y C son los estados aceptadores de M, tenemos que

$$L(M) = \{x \in \{0,1\}^* : n_0(x) \text{ es par o } n_1(x) \text{ es par}\}.$$

(2) Se tiene que $\Lambda(A) = \{A, B, C\} = ABC$, $\Lambda(B) = B$, $\Lambda(C) = C$, $\Lambda(D) = D$ y $\Lambda(E) = E$. Construimos entonces el autómata determinista M' equivalente a M. El estado inicial de M' es $\Lambda(A) = ABC$. Definimos ahora la función de transición δ' para M'.

$\delta'(ABC, 0) = \Lambda(C) \cup \Lambda(D) = CD$,
$\delta'(ABC, 1) = \Lambda(B) \cup \Lambda(E) = BE$,
$\delta'(CD, 0) = \Lambda(B) \cup \Lambda(C) = BC$,

$\delta'(CD, 1) = \Lambda(D) \cup \Lambda(E) = DE,$
$\delta'(BE, 0) = \Lambda(D) \cup \Lambda(E) = DE,$
$\delta'(BE, 1) = \Lambda(B) \cup \Lambda(C) = BC.$
$\delta'(BC, 0) = \Lambda(C) \cup \Lambda(D) = CD,$
$\delta'(BC, 1) = \Lambda(B) \cup \Lambda(E) = BE,$
$\delta'(DE, 0) = \Lambda(B) \cup \Lambda(E) = BE,$
$\delta'(DE, 1) = \Lambda(C) \cup \Lambda(D) = CD.$

Por tanto, los estados de M' son: ABC, CD, BE, BC y DE. Como B y C son los estados aceptadores de M, los estados aceptadores de M' son ABC, CD, BE y BC.

(3) Representamos a ABC por 0, a CD por 1, a BE por 2, a BC por 3 y a DE por 4. Podemos escribir entonces el siguiente programa en JAVA para simular el autómata M':

```
public boolean simular (String entrada)
{ int q = 0, i = 0;
  char c = entrada.charAt(0);
  while (c != '$')
  { switch(q)
        { case 0:
        if (c == '0') q = 1; else if (c == '1') q = 2;
        break;
        case 1:
        if (c == '0') q = 3; else if (c == '1') q = 4;
        break;
        case 2:
        if (c == '0') q = 4; else if (c == '1') q = 3;
        break;
        case 3:
        if (c == '0') q = 1; else if (c == '1') q = 2;
        break;
        case 4:
        if (c == '0') q = 2; else if (c == '1') q = 1;
        break; }
  c = entrada.charAt(++i); }
  if (q == 4) return false; else return true; }
```

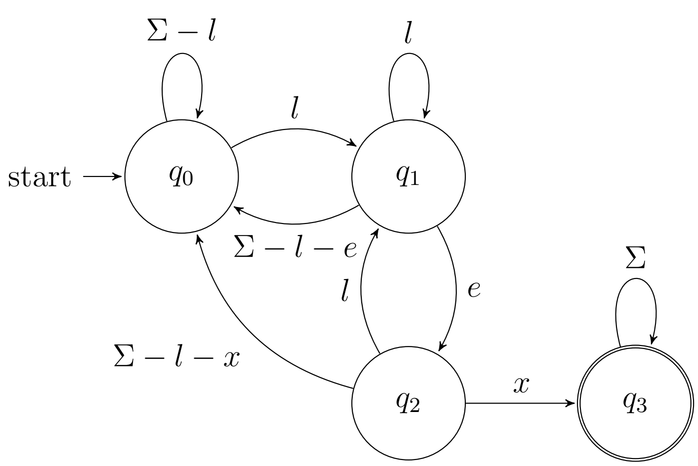
Problema 8. (a) Definir un autómata determinista que determine si un texto de caracteres contiene el patrón "lex".

(b) Definir un autómata indeterminista que determine si un texto de caracteres contiene el patrón "lex" o el patrón "ens2001".

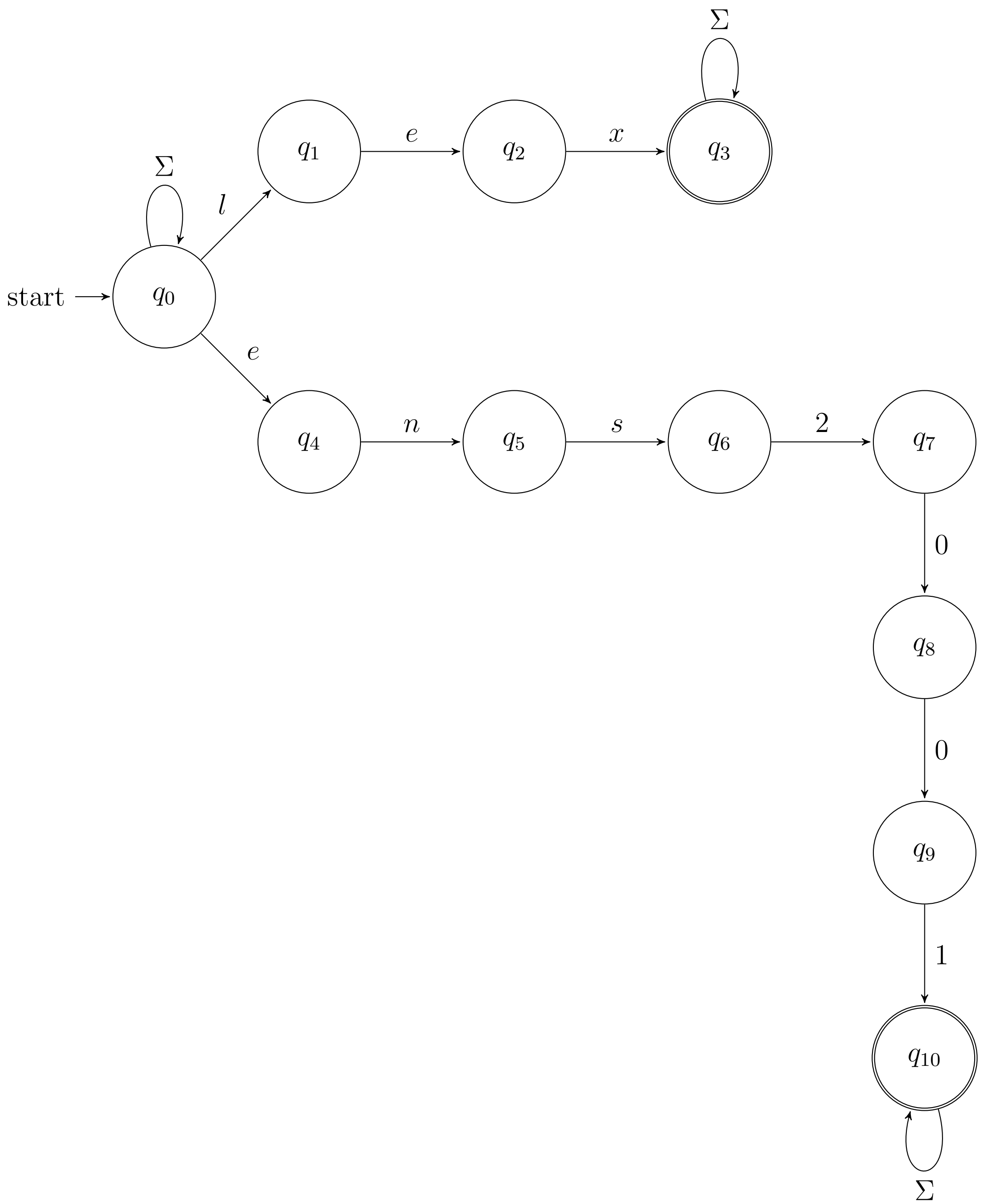
(c) Programar en Java el autómata determinista definido en (a).

Solución:

(a) Denotamos por Σ al conjunto de los caracteres ASCII. El siguiente autómata determinista determina entonces si un texto de caracteres contiene el patrón "lex".



(b) Denotamos por Σ al conjunto de los caracteres ASCII. El siguiente autómata indeterminista determina entonces si un texto de caracteres contiene el patrón "lex" o el patrón "ens2001".


start
q0
q1
q2
q3
q4
q5
q6
q7
q8
q9
q10
Σ
l
e
x
n
s
2
0
1

(c) Podemos escribir el siguiente programa en JAVA para simular el autómata determinista del apartado (a).

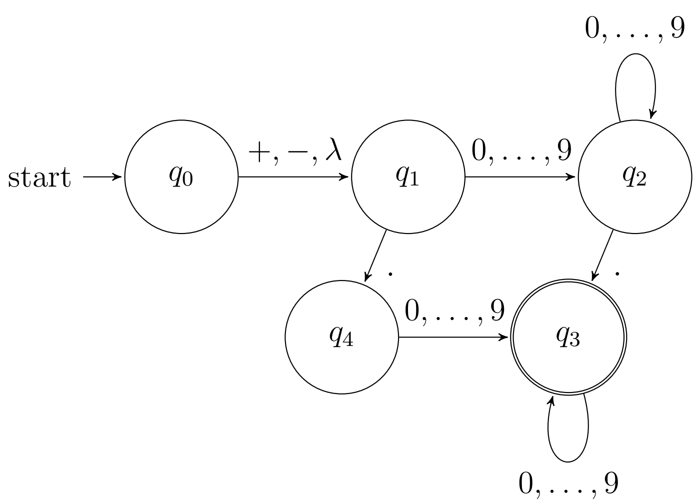
```
public boolean simular (String entrada)
{ int q = 0, i = 0;
  char c = entrada.charAt(0);
  while (c != '$')
  { switch(q)
        { case 0:
        if (c == 'l') q = 1;
        break;
        case 1:
        if (c == 'e') q = 2; else if ((c != 'e') && (c != 'l')) q = 0;
        break;
        case 2:
        if (c == 'x') return true; else if (c == 'l') q = 1; else q = 0;
        break; }
  c = entrada.charAt(++i); }
  return false; }
```

Problema 9. (a) Definir un autómata indeterminista para reconocer números decimales que contengan: (a) un signo + o − opcional; (b) una palabra de dígitos; (c) un punto decimal; (d) una segunda palabra de dígitos. Tanto la primera palabra de dígitos como la segunda pueden estar vacías, pero al menos una de las dos palabras no puede estar vacía.

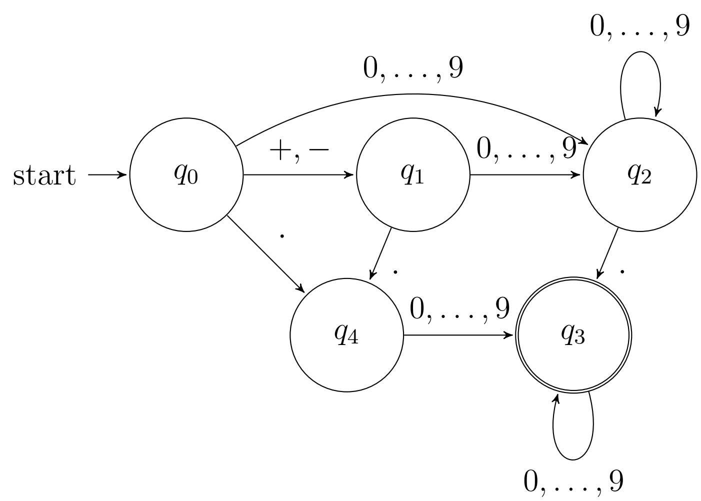
(b) Explicar, cómo utilizando el autómata del apartado (a), se puede escribir un programa en JAVA para reconocer números decimales (el tipo "float").

Solución:

(a) En primer lugar, definimos el siguiente autómata indeterminista M_1, en donde q_0 es el estado inicial, y q_3 es el único estado aceptador.



(b) Para escribir un programa en JAVA que reconozca los números decimales, en primer lugar hemos de eliminar el indeterminismo del autómata M_1 del apartado (a). Para ello, podemos utilizar el algoritmo visto en clase de teoría para transformar un autómata indeterminista en un autómata determinista equivalente. Sin embargo, en este caso podemos construir directamente el autómata determinista equivalente. Para ello, construimos el siguiente autómata M_2 equivalente a M_1, en el que eliminamos la transición con la λ.



Ahora, podemos definir el autómata determinista equivalente, simplemente añadiendo a este último autómata M_2 un estado de error, al cual llegaremos desde cualquier otro estado cuando entre un carácter distinto a los indicados en las transiciones del gráfico del autómata M_2. El programa asociado a este autómata determinista será entonces el programa buscado para reconocer los números decimales.

Problema 10. Explicar cómo diseñar un analizador léxico para reconocer las siguientes categorías sintácticas:

(i) identificadores formados por letras y dígitos de manera que el primer carácter es una letra,

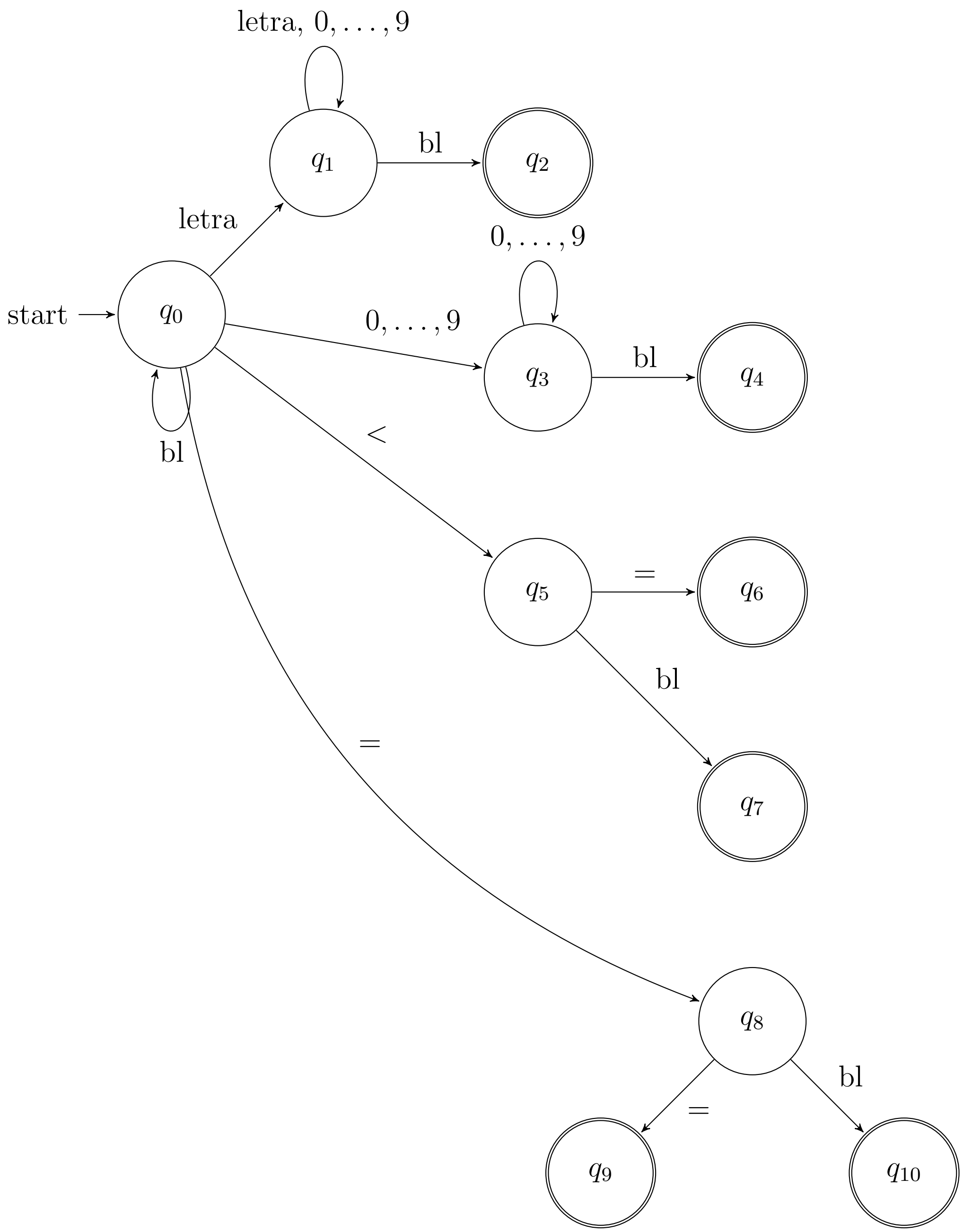
(ii) números enteros sin signo,

(iii) la asignación $=$,

(iv) los predicados $==$, $<$ y $<=$.

Solución:

En primer lugar, construimos el siguiente autómata indeterminista M para reconocer las categorías sintácticas indicadas, en el cual los estados aceptadores son q_2, q_4, q_6, q_7, q_9 y q_{10}. Representamos por bl al carácter blanco.



Por tanto, el estado q_2 reconoce un identificador formado por letras y dígitos, el estado q_4 reconoce un entero sin signo, el estado q_6 reconoce $<=$, el estado q_7 reconoce $<$, el estado q_9 reconoce $==$ y el estado q_{10} reconoce la asignación =. Se observa que los estados q_2, q_4, q_7 y q_{10} van adelantados un carácter, ya que dichos estados reconocen la categoría sintáctica al leer el símbolo siguiente a la categoría. Entonces, para escribir el analizador léxico, en primer lugar hemos de añadir un estado de error al autómata M, al cual llegaremos desde cualquier otro estado cuando entre un carácter distinto a los indicados en las transiciones del gráfico del autómata M. A continuación, programamos el autómata resultante, siguiendo el método visto en clase, con las siguientes modificaciones:

- Cuando se llegue a un estado aceptador, el programa imprimirá la categoría sintáctica reconocida y volverá al estado inicial.
- Cuando se llegue a q_2, a q_4, a q_7 o a q_{10}, el programa no leerá el siguiente símbolo de la entrada. Y sí lo leerá, cuando llegue a cualquier otro estado.

Problema 1. Definir autómatas deterministas con cuatro estados que reconozcan los siguientes lenguajes.

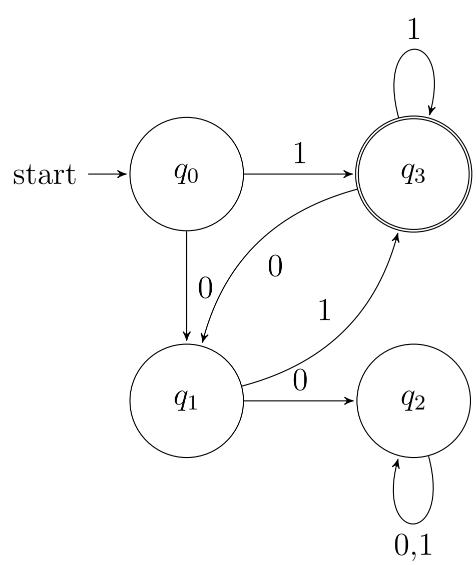
(a) $\{x \in \{0,1\}^* : x$ acaba en 1 y no contiene 00$\}$.

(b) $\{x \in \{0,1\}^* : x$ no contiene ni 00 ni 11$\}$.

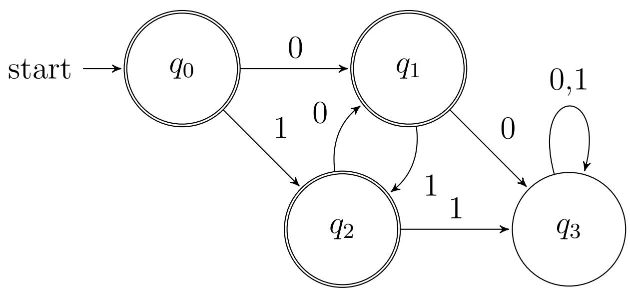
(c) $\{x \in \{0,1\}^* :$ cada 0 en x va inmediatamente precedido e inmediatamente seguido po un 1$\}$.

Solución:

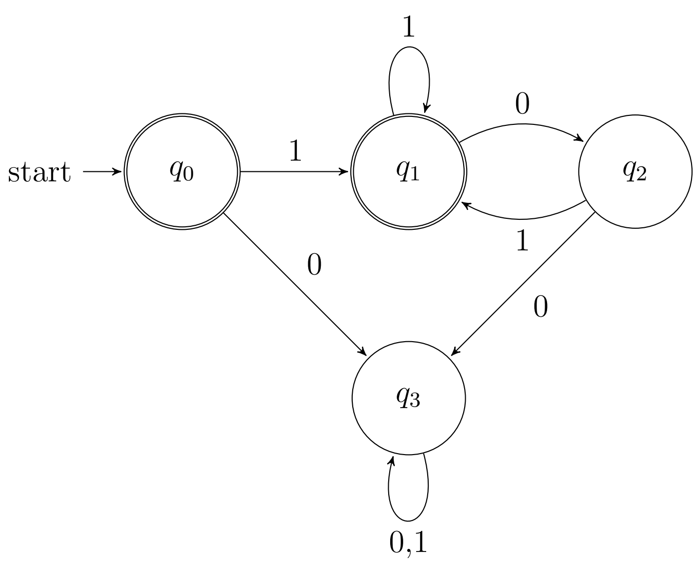
(a) Definimos el siguiente autómata determinista, que consta de los estados q_0, q_1, q_2 y q_3, donde q_0 es el estado inicial y q_3 es el único estado aceptador.



(b) Definimos el siguiente autómata determinista, que consta de los estados q_0, q_1, q_2 y q_3, donde q_0 es el estado inicial y q_0, q_1 y q_2 son los estados aceptadores.



(c) Definimos el siguiente autómata determinista, que consta de los estados q_0, q_1, q_2 y q_3, donde q_0 es el estado inicial y q_0 y q_1 son los estados aceptadores.

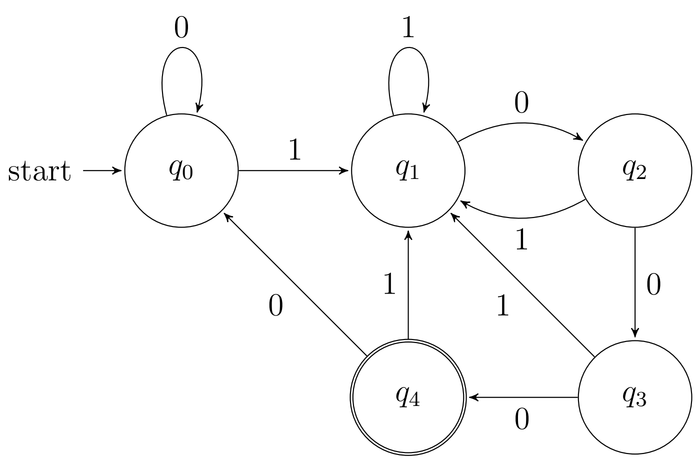


Problema 2. (a) Definir un autómata determinista que reconozca el lenguaje de las palabras de bits que acaban en 1000.

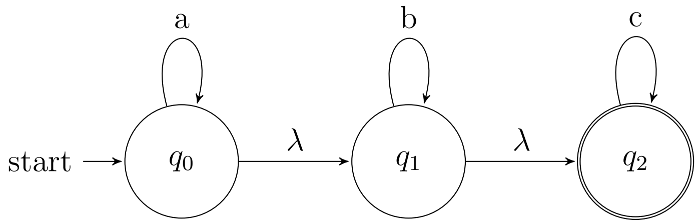
(b) Definir un autómata indeterminista que reconozca el lenguaje de las palabras con cero o más letras a, seguidas de cero o más letras b, seguidas de cero o más letras c .

Solución:

(a) Definimos el autómata por el siguiente gráfico, donde q_4 es el único estado aceptador.



(b) Definimos el siguiente autómata indeterminista, que consta de los estados q_0, q_1 y q_2, donde q_0 es el estado inicial y q_2 es el único estado aceptador.

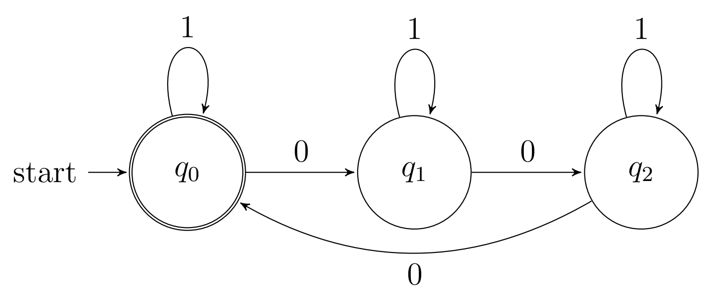

start
a
b
c
q_0
q_1
q_2
λ
λ

Problema 3. (a) Definir un autómata determinista M tal que $L(M) = \{x \in \{0,1\}^* : n_0(x) \text{ es un múltiplo de } 3\}$.

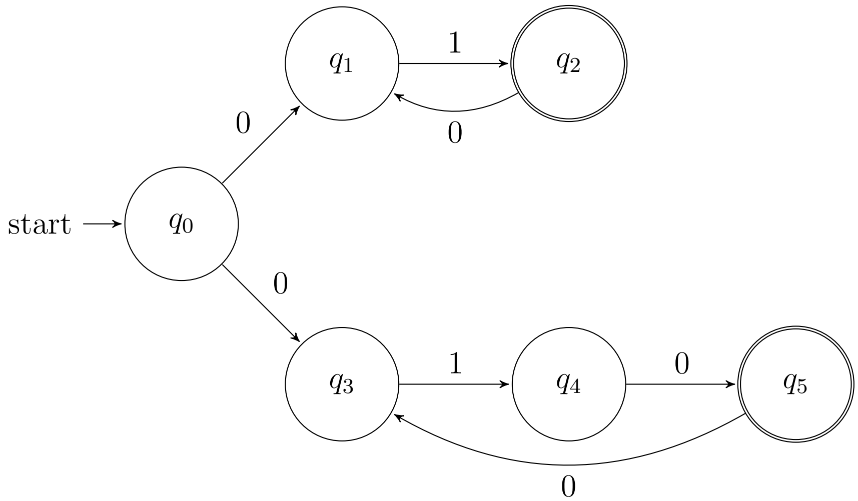
(b) Definir un autómata indeterminista M tal que $L(M) = \{(01)^n : n \geq 1\} \cup \{(010)^n : n \geq 1\}$, es decir, $L(M)$ es el lenguaje de las palabras formadas por 01 repetido una o más veces, o por 010 repetido una o más veces.

Solución:

(a) Definimos el siguiente autómata determinista, que consta de los estados q_0, q_1 y q_2, donde q_0 es el estado inicial y es asimismo el único estado aceptador.



(b) Definimos el siguiente autómata indeterminista, que consta de los estados q_0, q_1, q_2, q_3, q_4 y q_5, donde q_0 es el estado inicial y q_2 y q_5 son los estados aceptadores.



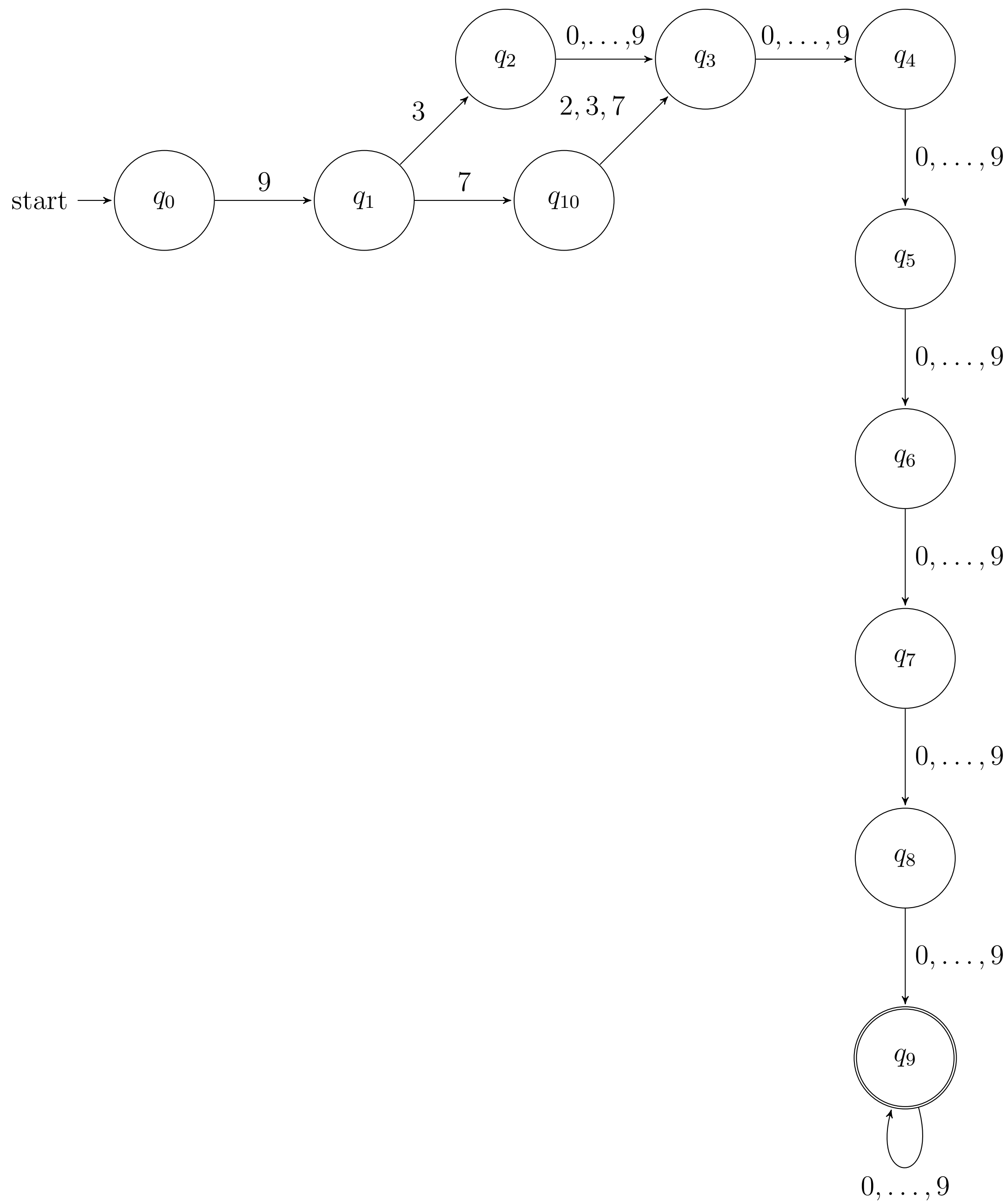
start
q_0
q_1
q_2
q_3
q_4
q_5
0
0
1
0
1
0
0

Problema 4. (a) Explicar cómo se puede construir un autómata indeterminista para reconocer los números de teléfono de las provincias de Catalunya.

(b) Explicar cómo a partir del l'autómata del apartado (a),se puede escibir un programa en JAVA para reconocer tales números

Solución:

(a) Recordemos que el prefijo para llamar a Girona es el 972, a Lleida es el 973 y a Tarragona es el 977. El siguiente autómata indeterminista, en donde q_9 es el único estado aceptador, reconoce los números de teléfono de Catalunya.


start
q_0
9
q_1
3
7
q_2
q_{10}
0,. . . ,9
2, 3, 7
q_3
0, . . . , 9
q_4
0, . . . , 9
q_5
0, . . . , 9
q_6
0, . . . , 9
q_7
0, . . . , 9
q_8
0, . . . , 9
q_9
0, . . . , 9

(b) Para escribir un programa en JAVA que reconozca los números de teléfono de Catalunya, en primer lugar hemos de eliminar el indeterminismo del autómata del apartado (a). Podemos hacer esto directamente añadiendo un estado de error al autómata, al cual llegaremos desde cualquier otro estado cuando entre un dígito distinto a los indicados en las transiciones del gráfico del autómata del apartado (a). El programa asociado a este autómata determinista será entonces el programa buscado para reconocer los números de teléfono de Catalunya.

Problema 5. Consideremos el autómata indeterminista $M = (\{q_0, q_1, q_2, q_3, q_4, q_5\}, \{\text{digito}\}, \Delta, q_0, \{q_5\})$ donde Δ está definida por la siguiente tabla:

q_0	digito	q_1
q_1	λ	q_2
q_2	λ	q_3
q_2	λ	q_5
q_3	digito	q_4
q_4	λ	q_3
q_4	λ	q_5

Se pide entonces:

(1) Calcular los λ-cierres de los estados de M.

(2) Utilizando el algoritmo visto en clase, transformar el autómata M en un autómata determinista equivalente.

Solución:

(1) Recordad que para todo estado q, el λ-cierre de q es el conjunto de todos los estados p para los cuales hay un cómputo en el autómata desde el estado q hasta el estado p sin leer ningún símbolo de la entrada. Tenemos entonces :

$\Lambda(q_0) = \{q_o\}$,

$\Lambda(q_1) = \{q_1, q_2, q_3, q_5\}$,

$\Lambda(q_2) = \{q_2, q_3, q_5\}$,

$\Lambda(q_3) = \{q_3\}$,

$\Lambda(q_4) = \{q_3, q_4, q_5\}$,

$\Lambda(q_5) = \{q_5\}$.

(2) El estado inicial del autómata determinista M' equivalente a M es $\Lambda(q_0) = \{q_o\}$. Calculamos ahora la función de transición δ' de M'. Obsérvese que el único símolo terminal es digito. Tenemos entonces:

$\delta'(\{q_0\}, \text{digito}) = \Lambda(q_1) = \{q_1, q_2, q_3, q_5\}$,

$\delta'(\{q_1, q_2, q_3, q_5\}, \text{digito}) = \Lambda(q_4) = \{q_3, q_4, q_5\}$,

$\delta'(\{q_3, q_4, q_5\}, \text{digito}) = \Lambda(q_4) = \{q_3, q_4, q_5\}$.

Como q_5 es el único estado aceptador del autómata M, los estados aceptadores de M' son los estados que contienen a q_5, es decir, $\{q_1, q_2, q_3, q_5\}$ y $\{q_3, q_4, q_5\}$. Para simplificar la notación, pongamos $p_0 = \{q_0\}, p_1 = \{q_1, q_2, q_3, q_5\}$ y $p_2 = \{q_3, q_4, q_5\}$. Se observa ahora que los estados p_1 y p_2 son equivalentes, ya que desde cualquiera de los dos estados p_1, p_2, nunca volvemos al único estado no aceptador p_0, y por tanto el lenguaje del autómata simplificado con un único estado aceptador es el mismo que el lenguaje del autómata con los dos estados aceptadores p_1 y p_2. Por tanto, obtenemos el autómata determinista formado por dos estados p_0 y p_1 donde p_0 es el estado inicial y p_1 es el único estado aceptador y en donde la función de transición está definida por $\delta'(p_0, \text{digito}) = p_1$ y $\delta'(p_1, \text{digito}) = p_1$. Por tanto, el autómata reconoce los números enteros sin signo.

Problema 6. Consideremos el autómata indeterminista $M = (\{P, Q, R, S\}, \{a, b\}, \Delta, P, \{P, Q\})$ donde Δ está definida por la siguiente tabla:

P	a	S
P	a	Q
P	λ	Q
Q	b	Q
Q	λ	R
R	b	P
S	a	S
S	b	R

Siguiendo el método visto en clase, trasformar el autómata M en un autómata determinista equivalente.

Solución:

Calculamos primero los cierres de los estados. Se tiene que $\Lambda(P) = PQR$, $\Lambda(Q) = QR$, $\Lambda(R) = R$ y $\Lambda(S) = S$. Construimos entonces el autómata determinista M' equivalente a M. El estado inicial de M' es $\Lambda(P) = PQR$. Construimos la función de transición δ' para M'.

$\delta'(PQR, a) = \Lambda(Q) \cup \Lambda(S) = QRS$,
$\delta'(PQR, b) = \Lambda(P) \cup \Lambda(Q) = PQR$,
$\delta'(QRS, a) = \Lambda(S) = S$,
$\delta'(QRS, b) = \Lambda(P) \cup \Lambda(Q) \cup \Lambda(R) = PQR$,
$\delta'(S, a) = \Lambda(S) = S$,
$\delta'(S, b) = \Lambda(R) = R$,
$\delta'(R, a) = \emptyset$,
$\delta'(R, b) = \Lambda(P) = PQR$,
$\delta'(\emptyset, 0) = \delta'(\emptyset, 1) = \emptyset$.

Por tanto, los estados de M' son: PQR, QRS, S, R y $\emptyset$. Como P y Q son los estados aceptadores de M, los estados aceptadores de M' son PQR y QRS.

Problema 7. Consideremos el autómata indeterminista $M = (\{A, B, C, D, E\}, \{0, 1\}, \Delta, A, \{B, C\})$ donde Δ está definida por la siguiente tabla:

A	λ	B
A	λ	C
B	0	D
B	1	B
C	0	C
C	1	E
D	0	B
D	1	D
E	0	E
E	1	C

Se pide entonces:

(1) Describir el lenguaje $L(M)$.

(2) Siguiendo el método visto en clase, transformar el autómata M en un autómata determinista equivalente.

(3) Programar en JAVA el autómata determinista obtenido en (2).

Solución:

(1) Se observa que el estado B reconoce las palabras de bits que tienen un número par de ceros, y el estado C reconoce las palabras de bits que tienen un número par de unos. Por tanto, como B y C son los estados aceptadores de M, tenemos que

$$L(M) = \{x \in \{0, 1\}^* : n_0(x) \text{ es par o } n_1(x) \text{ es par}\}.$$

(2) Se tiene que $\Lambda(A) = \{A, B, C\} = ABC$, $\Lambda(B) = B$, $\Lambda(C) = C$, $\Lambda(D) = D$ y $\Lambda(E) = E$. Construimos entonces el autómata determinista M' equivalente a M. El estado inicial de M' es $\Lambda(A) = ABC$. Definimos ahora la función de transición δ' para M'.

$\delta'(ABC, 0) = \Lambda(C) \cup \Lambda(D) = CD$,
$\delta'(ABC, 1) = \Lambda(B) \cup \Lambda(E) = BE$,
$\delta'(CD, 0) = \Lambda(B) \cup \Lambda(C) = BC$,

$\delta'(CD, 1) = \Lambda(D) \cup \Lambda(E) = DE,$
$\delta'(BE, 0) = \Lambda(D) \cup \Lambda(E) = DE,$
$\delta'(BE, 1) = \Lambda(B) \cup \Lambda(C) = BC.$
$\delta'(BC, 0) = \Lambda(C) \cup \Lambda(D) = CD,$
$\delta'(BC, 1) = \Lambda(B) \cup \Lambda(E) = BE,$
$\delta'(DE, 0) = \Lambda(B) \cup \Lambda(E) = BE,$
$\delta'(DE, 1) = \Lambda(C) \cup \Lambda(D) = CD.$

Por tanto, los estados de M' son: ABC, CD, BE, BC y DE. Como B y C son los estados aceptadores de M, los estados aceptadores de M' son ABC, CD, BE y BC.

(3) Representamos a ABC por 0, a CD por 1, a BE por 2, a BC por 3 y a DE por 4. Podemos escribir entonces el siguiente programa en JAVA para simular el autómata M':

```
public boolean simular (String entrada)
{ int q = 0, i = 0;
  char c = entrada.charAt(0);
  while (c != '$')
  { switch(q)
        { case 0:
        if (c == '0') q = 1; else if (c == '1') q = 2;
        break;
        case 1:
        if (c == '0') q = 3; else if (c == '1') q = 4;
        break;
        case 2:
        if (c == '0') q = 4; else if (c == '1') q = 3;
        break;
        case 3:
        if (c == '0') q = 1; else if (c == '1') q = 2;
        break;
        case 4:
        if (c == '0') q = 2; else if (c == '1') q = 1;
        break; }
  c = entrada.charAt(++i); }
  if (q == 4) return false; else return true; }
```

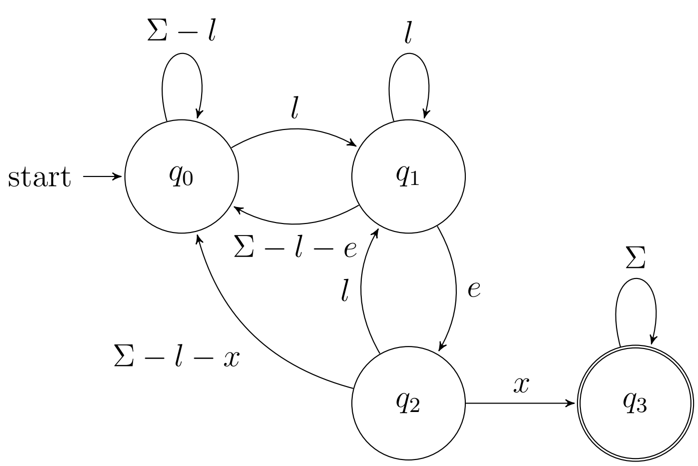
Problema 8. (a) Definir un autómata determinista que determine si un texto de caracteres contiene el patrón "lex".

(b) Definir un autómata indeterminista que determine si un texto de caracteres contiene el patrón "lex" o el patrón "ens2001".

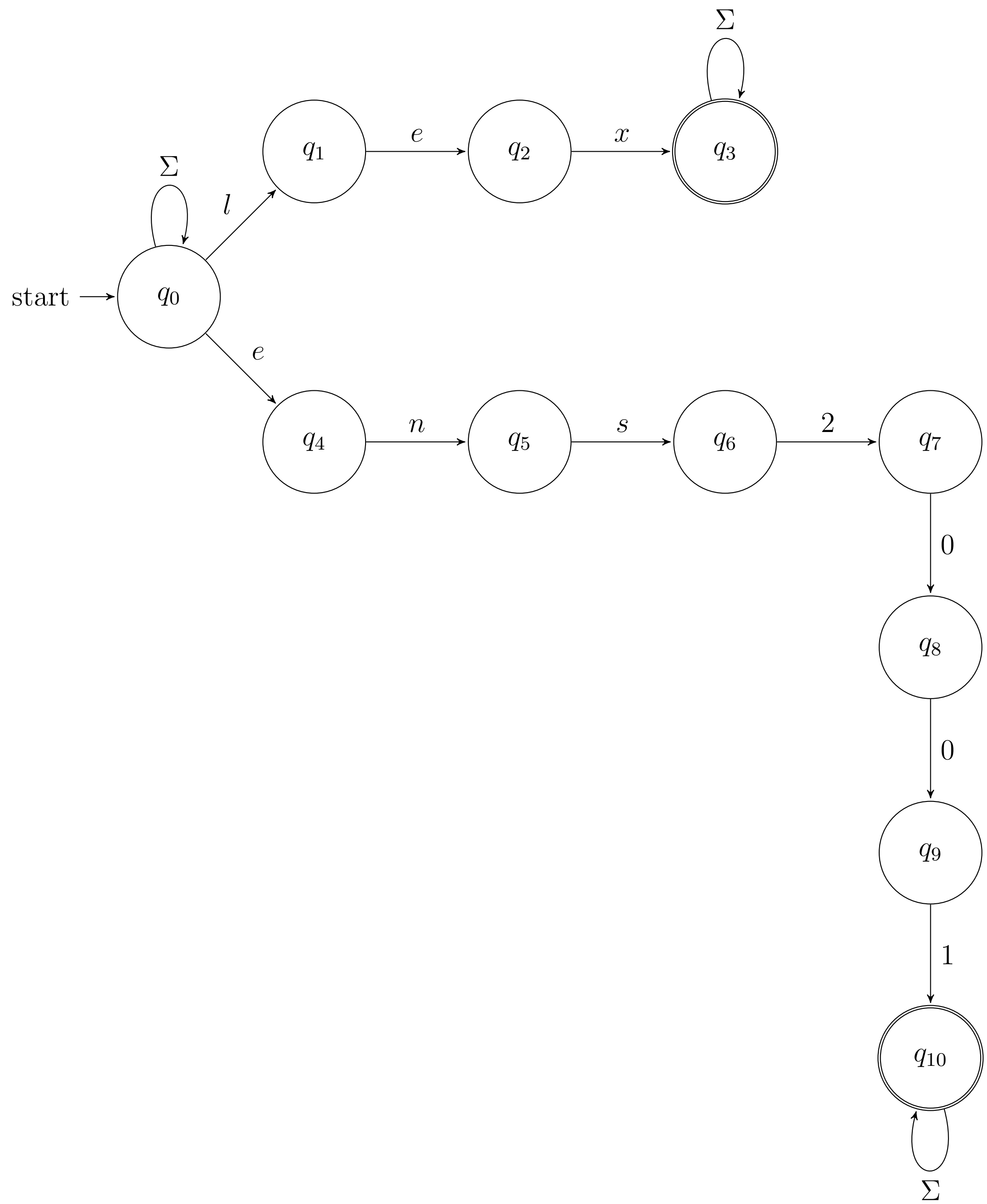
(c) Programar en Java el autómata determinista definido en (a).

Solución:

(a) Denotamos por Σ al conjunto de los caracteres ASCII. El siguiente autómata determinista determina entonces si un texto de caracteres contiene el patrón "lex".



(b) Denotamos por Σ al conjunto de los caracteres ASCII. El siguiente autómata indeterminista determina entonces si un texto de caracteres contiene el patrón "lex" o el patrón "ens2001".


start
q_0
q_1
q_2
q_3
q_4
q_5
q_6
q_7
q_8
q_9
q_{10}
Σ
l
e
x
n
s
2
0
1

(c) Podemos escribir el siguiente programa en JAVA para simular el autómata determinista del apartado (a).

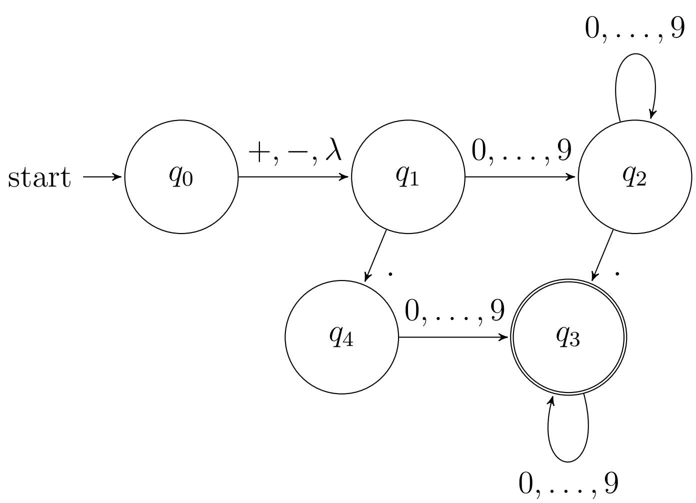
```
public boolean simular (String entrada)
{ int q = 0, i = 0;
  char c = entrada.charAt(0);
  while (c != '$')
  { switch(q)
        { case 0:
        if (c == 'l') q = 1;
        break;
        case 1:
        if (c == 'e') q = 2; else if ((c != 'e') && (c != 'l')) q = 0;
        break;
        case 2:
        if (c == 'x') return true; else if (c == 'l') q = 1; else q = 0;
        break; }
  c = entrada.charAt(++i); }
  return false; }
```

Problema 9. (a) Definir un autómata indeterminista para reconocer números decimales que contengan: (a) un signo + o − opcional; (b) una palabra de dígitos; (c) un punto decimal; (d) una segunda palabra de dígitos. Tanto la primera palabra de dígitos como la segunda pueden estar vacías, pero al menos una de las dos palabras no puede estar vacía.

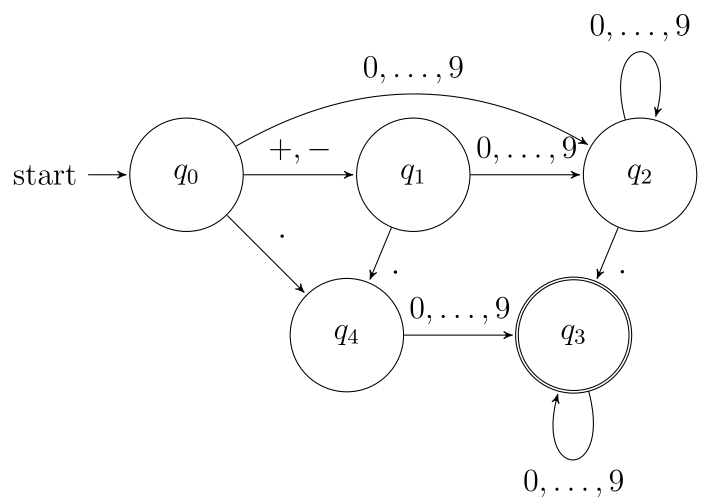
(b) Explicar, cómo utilizando el autómata del apartado (a), se puede escribir un programa en JAVA para reconocer números decimales (el tipo "float").

Solución:

(a) En primer lugar, definimos el siguiente autómata indeterminista M_1, en donde q_0 es el estado inicial, y q_3 es el único estado aceptador.



(b) Para escribir un programa en JAVA que reconozca los números decimales, en primer lugar hemos de eliminar el indeterminismo del autómata M_1 del apartado (a). Para ello, podemos utilizar el algoritmo visto en clase de teoría para transformar un autómata indeterminista en un autómata determinista equivalente. Sin embargo, en este caso podemos construir directamente el autómata determinista equivalente. Para ello, construimos el siguiente autómata M_2 equivalente a M_1, en el que eliminamos la transición con la λ.



Ahora, podemos definir el autómata determinista equivalente, simplemente añadiendo a este último autómata M_2 un estado de error, al cual llegaremos desde cualquier otro estado cuando entre un carácter distinto a los indicados en las transiciones del gráfico del autómata M_2. El programa asociado a este autómata determinista será entonces el programa buscado para reconocer los números decimales.

Problema 10. Explicar cómo diseñar un analizador léxico para reconocer las siguientes categorías sintácticas:

(i) identificadores formados por letras y dígitos de manera que el primer carácter es una letra,

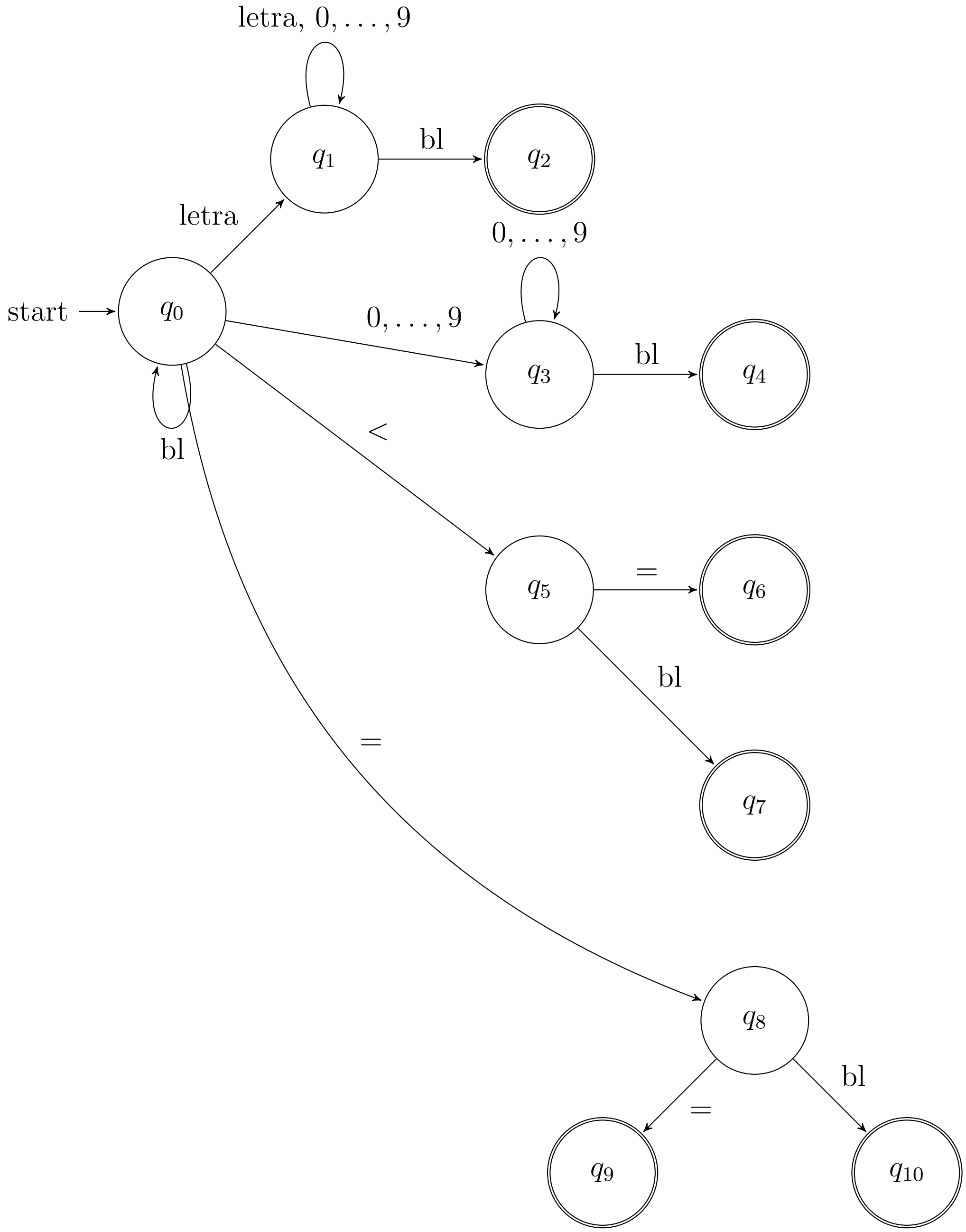
(ii) números enteros sin signo,

(iii) la asignación =,

(iv) los predicados ==, $<$ y $<=$.

Solución:

En primer lugar, construimos el siguiente autómata indeterminista M para reconocer las categorías sintácticas indicadas, en el cual los estados aceptadores son q_2, q_4, q_6, q_7, q_9 y q_{10}. Representamos por bl al carácter blanco.



Por tanto, el estado q_2 reconoce un identificador formado por letras y dígitos, el estado q_4 reconoce un entero sin signo, el estado q_6 reconoce $<=$, el estado q_7 reconoce $<$, el estado q_9 reconoce $==$ y el estado q_{10} reconoce la asignación =. Se observa que los estados q_2, q_4, q_7 y q_{10} van adelantados un carácter, ya que dichos estados reconocen la categoría sintáctica al leer el símbolo siguiente a la categoría. Entonces, para escribir el analizador léxico, en primer lugar hemos de añadir un estado de error al autómata M, al cual llegaremos desde cualquier otro estado cuando entre un carácter distinto a los indicados en las transiciones del gráfico del autómata M. A continuación, programamos el autómata resultante, siguiendo el método visto en clase, con las siguientes modificaciones:

- Cuando se llegue a un estado aceptador, el programa imprimirá la categoría sintáctica reconocida y volverá al estado inicial.
- Cuando se llegue a q_2, a q_4, a q_7 o a q_{10}, el programa no leerá el siguiente símbolo de la entrada. Y sí lo leerá, cuando llegue a cualquier otro estado.

LÒGICA I LLENGUATGES

CURSO 2020-21

TERCERA PRUEBA PARCIAL DE PROBLEMAS

(a) Definir un autómata determinista M tal que $L(M) = \{x \in \{a, b, c\}^* : n_a(x) + n_b(x) = 3\}$ y describir $L(M)$ mediante una expresión regular.

(2 puntos)

(b) Consideremos el autómata indeterminista $M = (\{A, B, C, D, E\}, \{0, 1\}, \Delta, A, \{E\})$ donde Δ está definida por la siguiente tabla:

A	1	B
A	1	C
A	0	E
B	1	A
C	1	D
D	0	A

Se pide entonces:

(1) Siguiendo el método visto en clase, transformar el autómata M en un autómata determinista equivalente. (5 puntos)

(2) Programar en JAVA el autómata determinista obtenido en (1).

(3 puntos)

LÒGICA I LLENGUATGES

CURSO 2021-22

TERCERA PRUEBA PARCIAL (Grupo A)

(a) Construir un autómata determinista M que reconozca los números de teléfono de 9 cifras que comiencen por 0600 y cuya última cifra sea par.

(3 puntos)

(b) Consideremos el autómata indeterminista $M = (\{A, B, C, D, \}, \{0, 1\}, \Delta, A, \{A, D\})$ donde Δ está definida por la siguiente tabla:

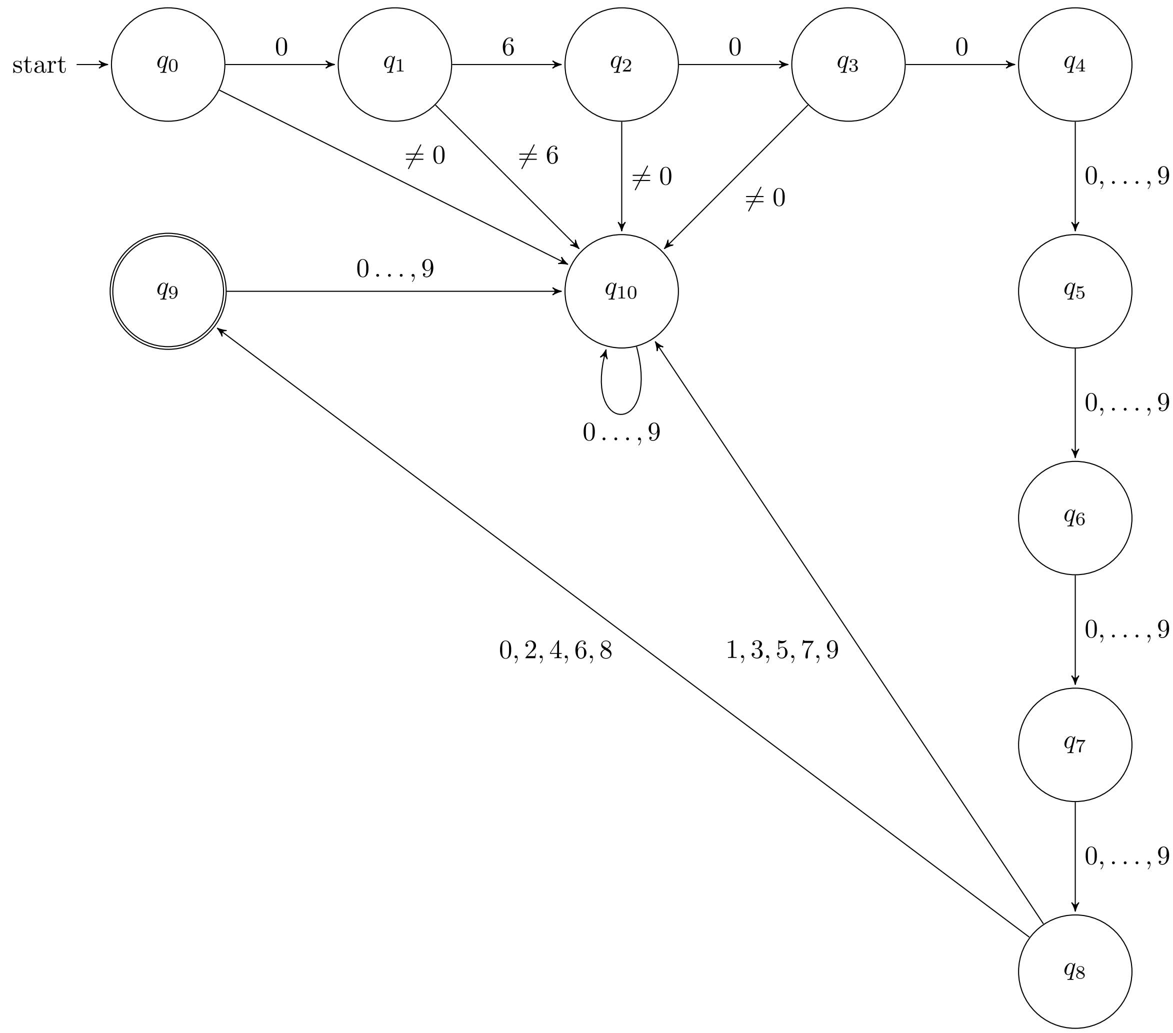
A	1	B
A	λ	C
B	0	B
B	0	C
B	1	C
C	0	A
C	λ	D
D	0	B
D	1	B

Se pide entonces:

(1) Siguiendo el método visto en clase, transformar el autómata M en un autómata determinista equivalente. (4,5 puntos)

(2) Programar en JAVA o en C el autómata determinista obtenido en (1). (2,5 puntos)

Solución: (a) El siguiente autómata determinista, en donde q_9 es el único estado aceptador, reconoce los números de teléfono indicados.



(b) (1) Calculamos primero los cierres de los estados. Se tiene que $\Lambda(A) = ACD$, $\Lambda(B) = B$, $\Lambda(C) = CD$ y $\Lambda(D) = D$. Construimos entonces el autómata determinista M' equivalente a M. El estado inicial de M' es $\Lambda(A) = ACD$. Construimos la función de transición δ' para M'.

$\delta'(ACD, 0) = \Lambda(A) \cup \Lambda(B) = ABCD$,

$\delta'(ACD, 1) = \Lambda(B) = B$,

$\delta'(ABCD, 0) = \Lambda(A) \cup \Lambda(B) \cup \Lambda(C) = ABCD$,

$\delta'(ABCD, 1) = \Lambda(B) \cup \Lambda(C) = BCD$,

$\delta'(B, 0) = \Lambda(B) \cup \Lambda(C) = BCD$,

$\delta'(B, 1) = \Lambda(C) = CD,$

$\delta'(BCD, 0) = \Lambda(A) \cup \Lambda(B) \cup \Lambda(C) = ABCD,$

$\delta'(BCD, 1) = \Lambda(B) \cup \Lambda(C) = BCD,$

$\delta'(CD, 0) = \Lambda(A) \cup \Lambda(B) = ABCD,$

$\delta'(CD, 1) = \Lambda(B) = B,$

Por tanto, los estados de M' son: $ACD, ABCD, B, BCD$, y CD. Como A y D son los estados aceptadores de M, los estados aceptadores de M' son ACD, $ABCD$, BCD y CD.

(2) Representamos al estado ACD por 0, al estado $ABCD$ por 1, al estado B por 2, al estado BCD por 3 y al estado CD por 4. Podemos escribir entonces el siguiente programa en JAVA para simular el autómata M':

```
public boolean simular (String entrada)
{ int q = 0, i = 0;
  char c = entrada.charAt(0);
  while (c != '$')
  { switch(q)
        { case 0:
        if (c == '0') q = 1; else if (c == '1') q = 2;
        break;
        case 1:
        if (c == '1') q = 3;
        break;
        case 2:
        if (c == '0') q = 3; else if (c == '1') q = 4;
        break;
        case 3:
        if (c == '0') q = 1;
        break;
        case 4:
        if (c == '0') q = 1; else if (c == '1') q = 2;
        break;}
  c = entrada.charAt(++i); }
  if (q == 2) return false; else return true; }
```

LÒGICA I LLENGUATGES

CURSO 2021-22

TERCERA PRUEBA PARCIAL (Grupo B)

(a) (1) Construir un autómata determinista M tal que $L(M) = \{x \in \{0,1\}^* : x \text{ acaba en } 00 \text{ o en } 11\ \}$.

(2) Construir un autómata indeterminista M tal que $L(M) = L(\alpha)$ donde α es la expresión regular $a^*b^*c^*$.

(3 puntos)

(b) Consideremos el autómata indeterminista $M = (\{A, B, C, D, E, F, G\}, \{0, 1\}, \Delta, A, \{C, D, F\})$ donde Δ está definida por la siguiente tabla:

A	λ	B
A	λ	C
B	0	D
B	0	E
C	1	C
C	1	G
E	1	F
F	0	E
G	0	C

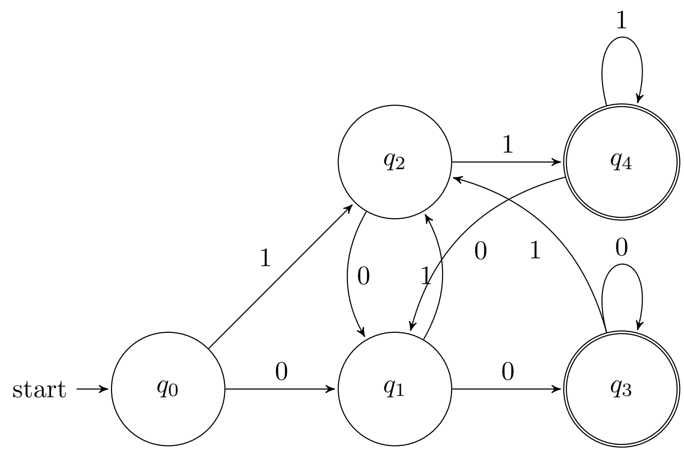
Se pide entonces:

(1) Describir el lenguaje $L(M)$ mediante una expresión regular. (1 punto)

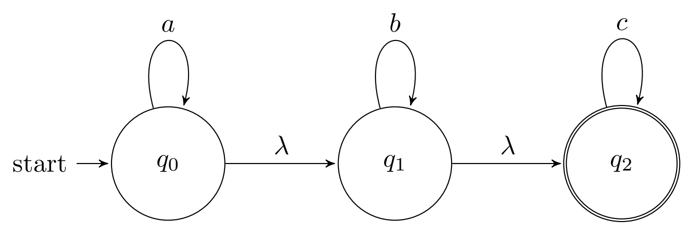
(2) Siguiendo el método visto en clase, transformar el autómata M en un autómata determinista equivalente. (4 puntos)

(3) Programar en JAVA o en C el autómata determinista obtenido en (2). (2 puntos)

Solución: (a) (1) Construimos el siguiente autómata determinista:



(2) Construimos el siguiente autómata indeterminista:



(b) (1) Tenemos que $L(M) = L(\alpha)$ donde $\alpha = 0 \cup (01)(01)^* \cup (1 \cup 10)^*$.

(2) Se tiene que $\Lambda(A) = ABC$, $\Lambda(B) = B$, $\Lambda(C) = C$, $\Lambda(D) = D$, $\Lambda(E) = E$, $\Lambda(F) = F$ y $\Lambda(G) = G$. Construimos entonces el autómata determinista M' equivalente a M. El estado inicial de M' es $\Lambda(A) = ABC$. Construimos la función de transición δ' para M'.

$\delta'(ABC, 0) = \Lambda(D) \cup \Lambda(E) = DE$,
$\delta'(ABC, 1) = \Lambda(C) \cup \Lambda(G) = CG$,
$\delta'(DE, 0) = \emptyset$,
$\delta'(DE, 1) = \Lambda(F) = F$,
$\delta'(CG, 0) = \Lambda(C) = C$,
$\delta'(CG, 1) = \Lambda(C) \cup \Lambda(G) = CG$,
$\delta'(F, 0) = \Lambda(E) = E$,
$\delta'(F, 1) = \emptyset$,

$\delta'(C,0) = \emptyset$,
$\delta'(C,1) = \Lambda(C) \cup \Lambda(G) = CG$,
$\delta'(E,0) = \emptyset$,
$\delta'(E,1) = \Lambda(F) = F$,
$\delta'(\emptyset,0) = \delta'(\emptyset,1) = \emptyset$.

Por tanto, los estados de M' son: ABC, DE, CG, E, F, C y $\emptyset$. Como C, D y F son los estados aceptadores de M, los estados aceptadores de M' son ABC, DE, CG, F y C.

(3) Representamos al estado ABC por 0, al estado DE por 1, al estado CG por 2, al estado F por 3, al estado C por 4, y al estado E por 5. Como $\emptyset$ es un estado de error, no hace falta representarlo. Podemos escribir entonces el siguiente programa en JAVA para simular el autómata M':

```
public boolean simular (String entrada)
{ int q = 0, i = 0;
  char c = entrada.charAt(0);
  while (c != ‘$’)
  { switch(q)
        { case 0:
        if (c == ‘0’) q = 1; else if (c == ’1’) q = 2;
        break;
        case 1:
        if (c == ‘0’) return false; else if (c == ‘1’) q = 3;
        break;
        case 2:
        if (c == ‘0’) q = 4;
        break;
        case 3:
        if (c == ‘0’) q = 5; else if (c == ’1’) return false;
        break;
        case 4:
        if (c == ‘1’) q = 2; else if (c == ’0’) return false;
        break;
        case 5:
        if (c == ‘0’) return false; else if (c == ’1’) q = 3;
        break;}
  c = entrada.charAt(++i); }
  if (q == 5) return false; else return true; }
```