UNIVERSIDAD DE CÓRDOBA
ESCUELA POLITÉCNICA SUPERIOR
DEPARTAMENTO DE INFORMÁTICA Y ANÁLISIS NUMÉRICO

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

SEGUNDO CURSO, SEGUNDO CUATRIMESTRE

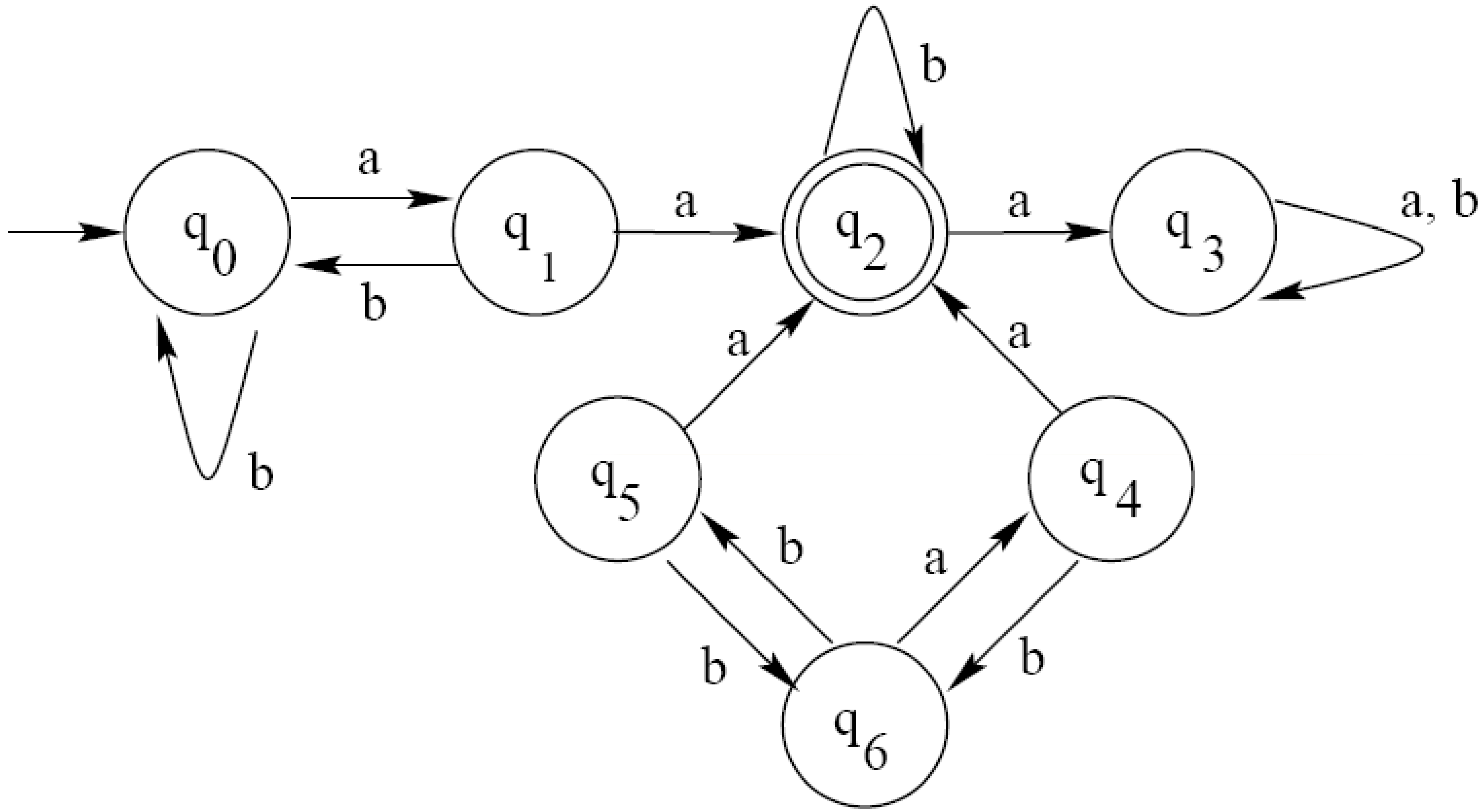
TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES

Tema 6.- Autómatas finitos y máquinas secuenciales

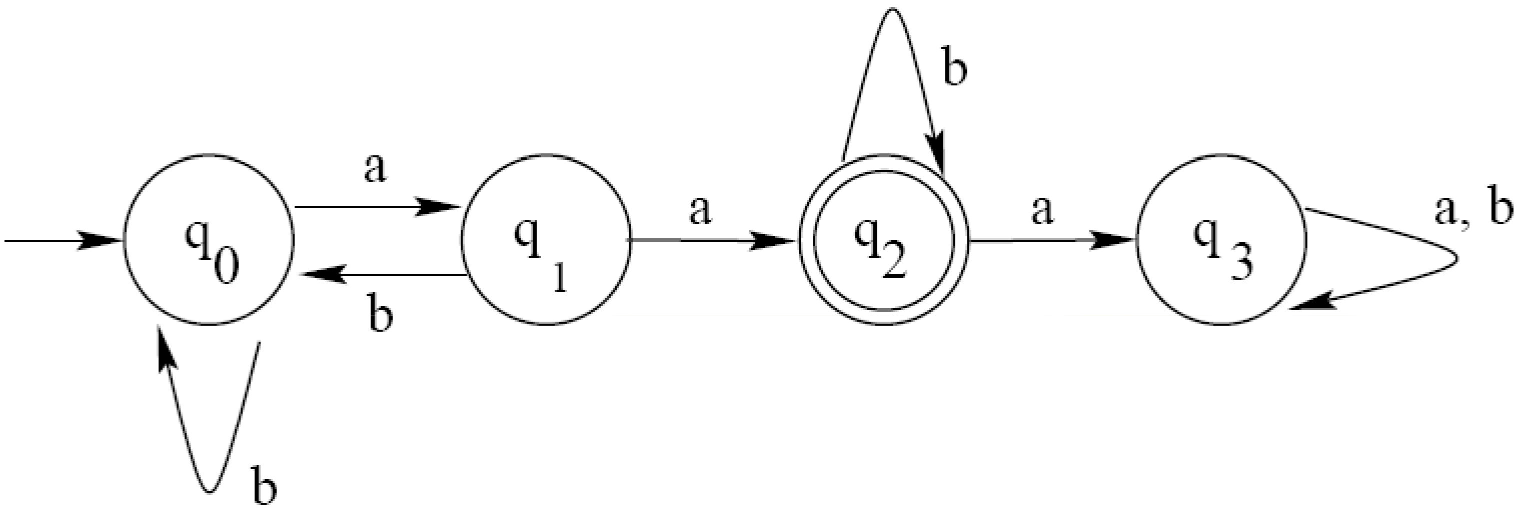
DIAN
DPTO. DE INFORMÁTICA Y ANÁLISIS NUMÉRICO

Algoritmo para la obtención de los estados accesibles

```
inicio
    Accesibles ← {q0} y q0 no marcado
    mientras haya un estado no marcado q ∈ Accesibles hacer
        Marcar el estado q
        para cada σ ∈ Σ hacer
            si  δ(q, σ) = q' ∉ Accesibles
                entonces Accesibles ← Accesibles ∪ {q'}
                         y q' no marcado
            fin si
        fin para
    fin mientras
fin
```



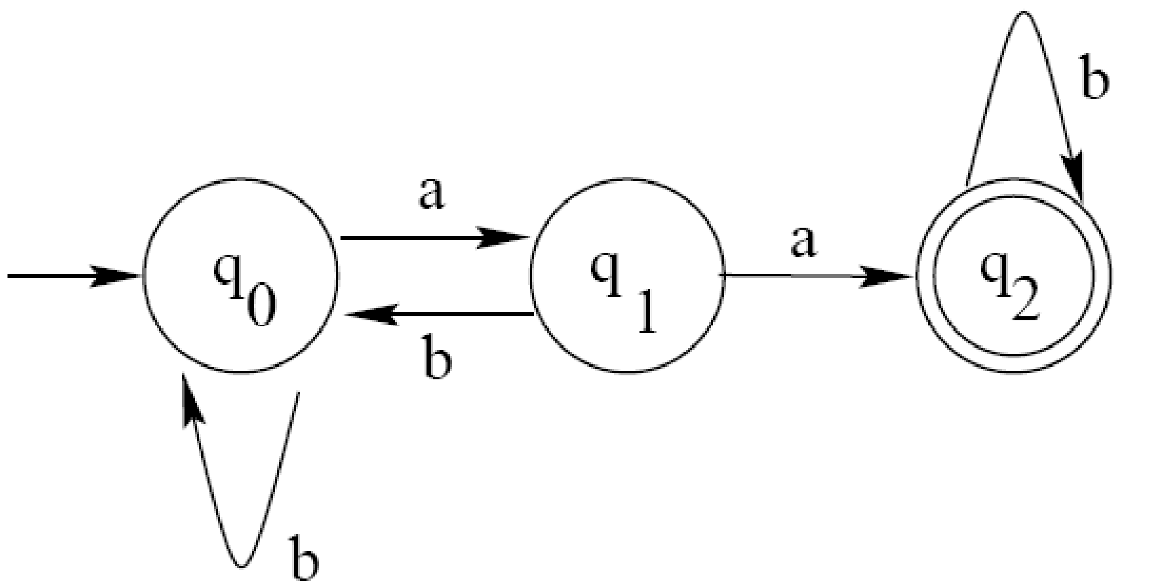
Autómata **con** estados inútiles



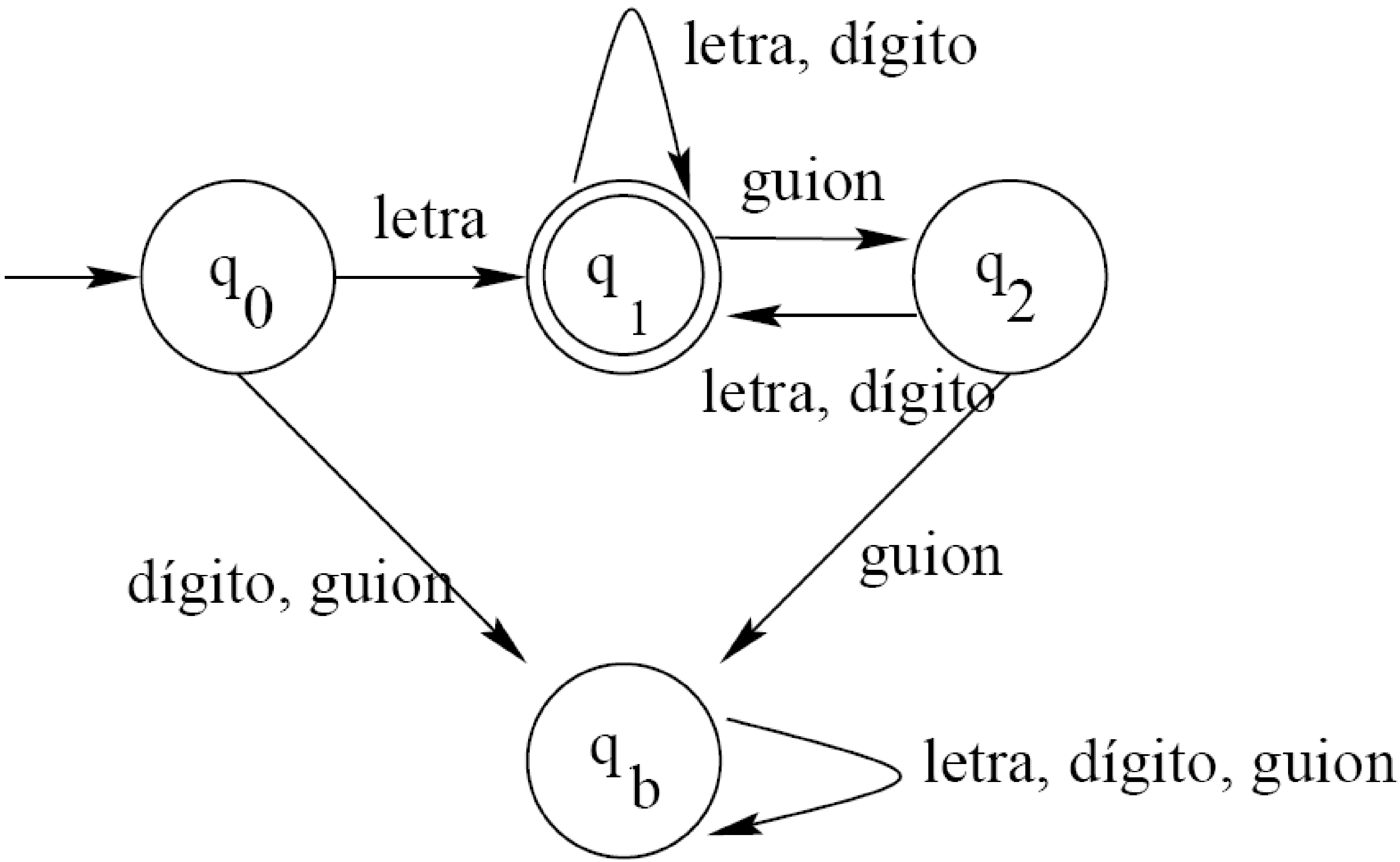
Autómata conexo pero con estados **no** finalizables

Algoritmo para la obtención de los estados finalizables

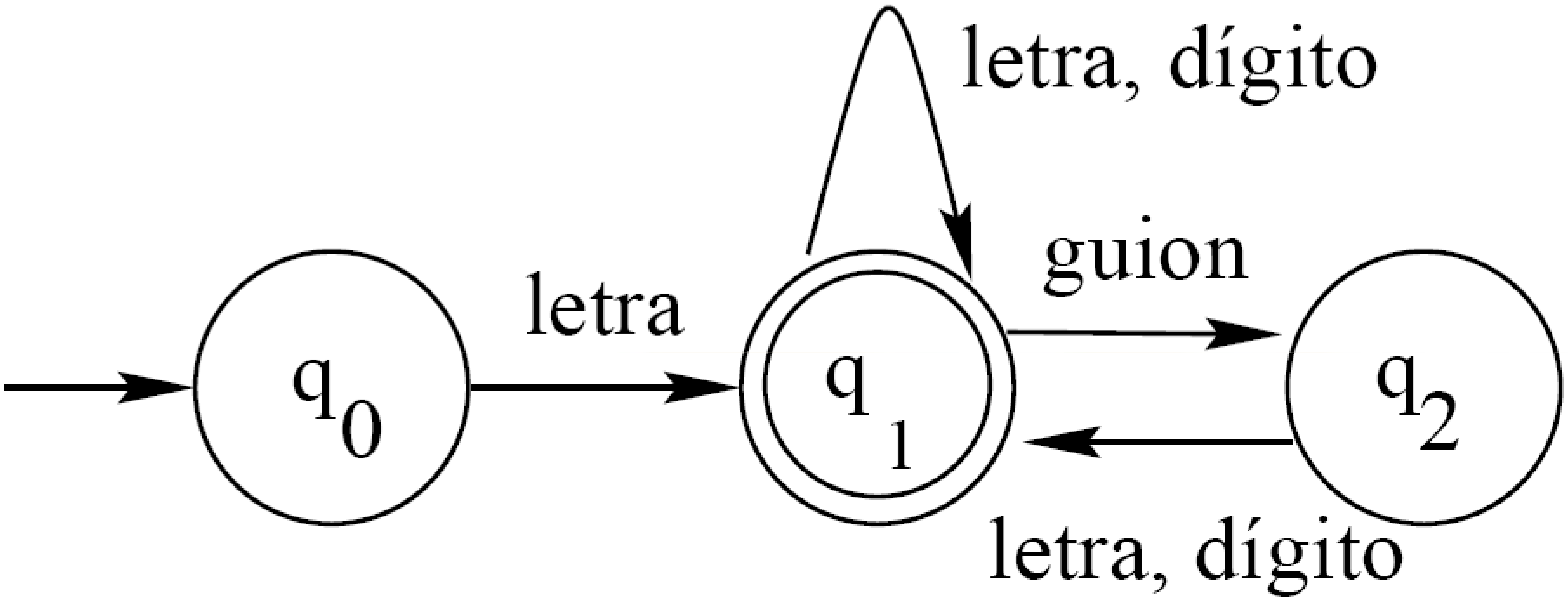
```
inicio
    Viejo ← ∅
    Nuevo ← F
    mientras (Viejo ≠ Nuevo) hacer
        Viejo ← Nuevo
        para cada q ∉ Viejo hacer
            si ∃ σ ∈ Σ, δ(q, σ) ∈ Nuevo
                entonces Nuevo ← Nuevo ∪ {q}
            fin si
        fin para
    fin mientras
    Finalizables ← Nuevo
fin
```



Autómata **sin** estados inútiles



Autómata **con** un estado inútil



Autómata **sin** estados inútiles

Algoritmo para obtener el "Autómata Cociente"

inicio

  $p_0 \leftarrow Q - F$

  $p_1 \leftarrow F$

  Nuevo $\leftarrow \{p_0, p_1\}$ *y* p_0 *y* p_1 *no marcados*

  mientras *haya un estado* $\mathbf{p} \in$ *Nuevo no marcado* ***hacer***

    Marcar a p

    para *cada* $\sigma \in \Sigma$ ***hacer***

      Dividir ***p*** *en subconjuntos tales que* q_i *y* q_j *estarán*

      en el mismo subconjunto si $\delta(q_i, \sigma)$ *y* $\delta(q_j, \sigma)$

      pertenecen al mismo subconjunto de Nuevo

    fin para

    si *se ha dividido* ***p*** *en subconjuntos*

      entonces

        Sustituir ***p*** *en Nuevo por los subconjuntos obtenidos*

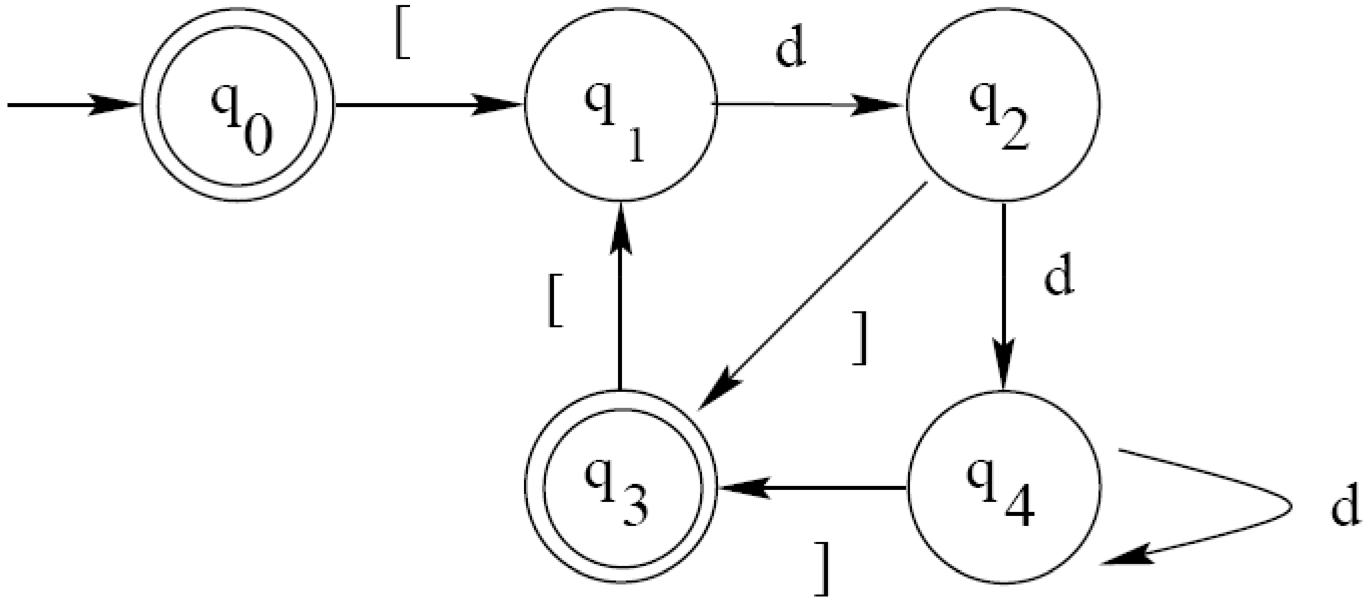
        Desmarcar a todos los estados de Nuevo

    fin si

  fin mientras

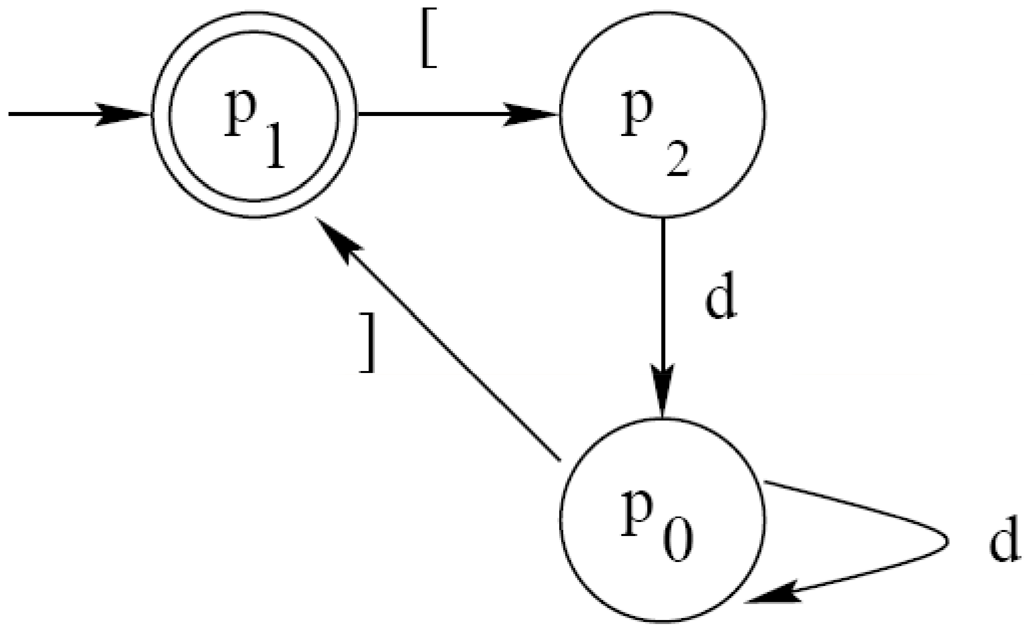
fin

Minimización de un autómata



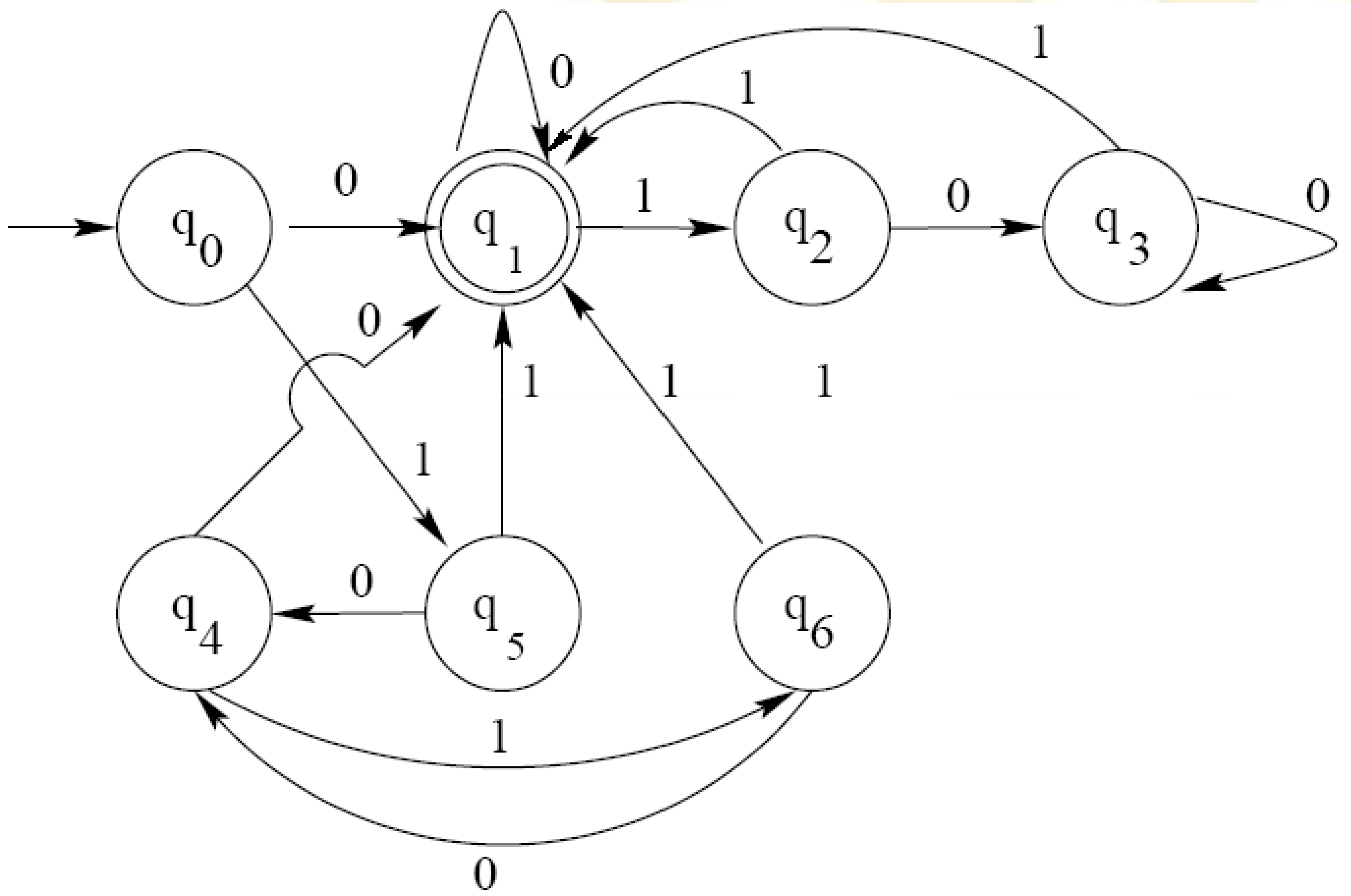
Autómata **no** minimizado

Minimización de un autómata



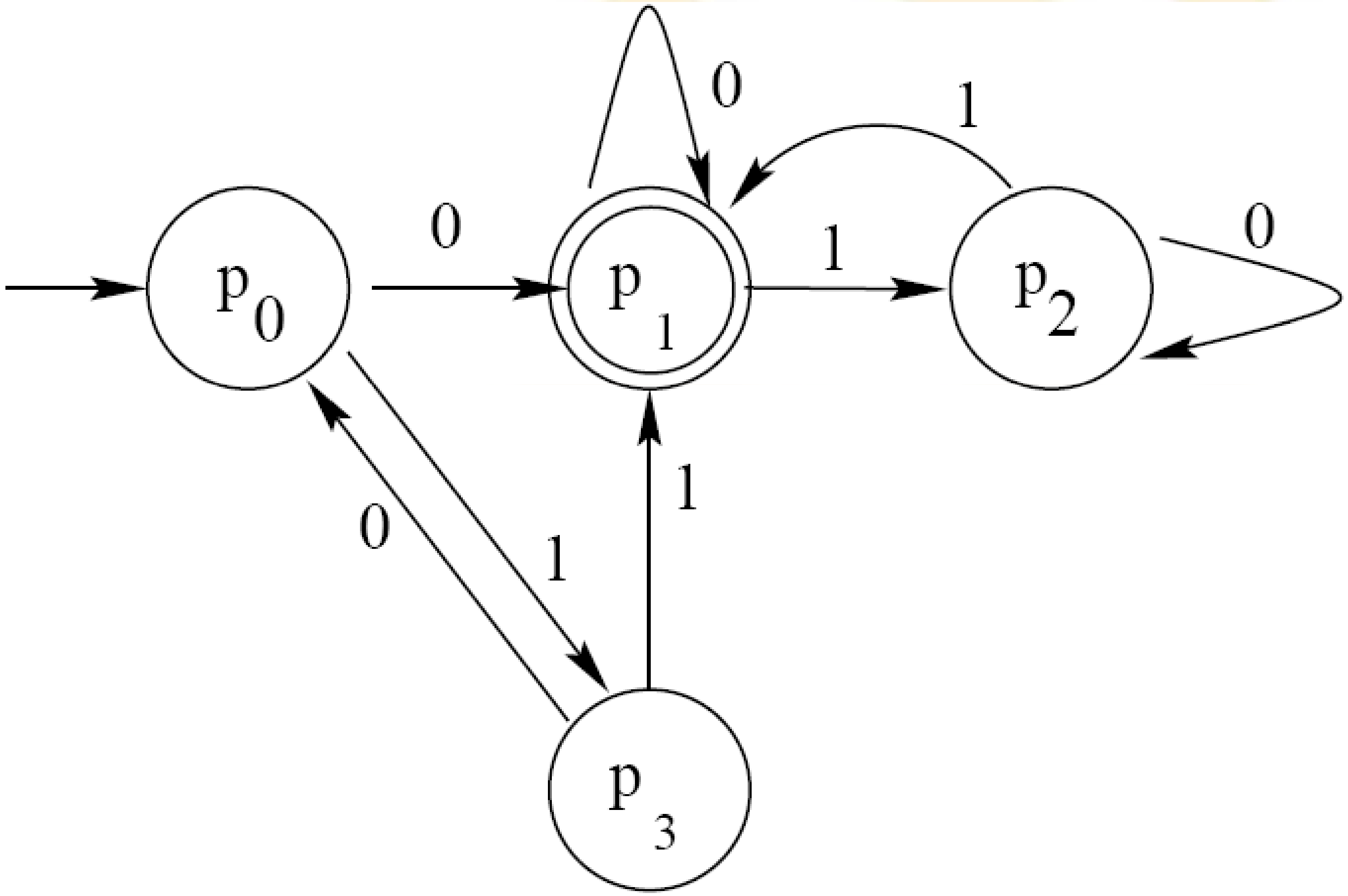
Autómata **minimizado**

Minimización de un autómata

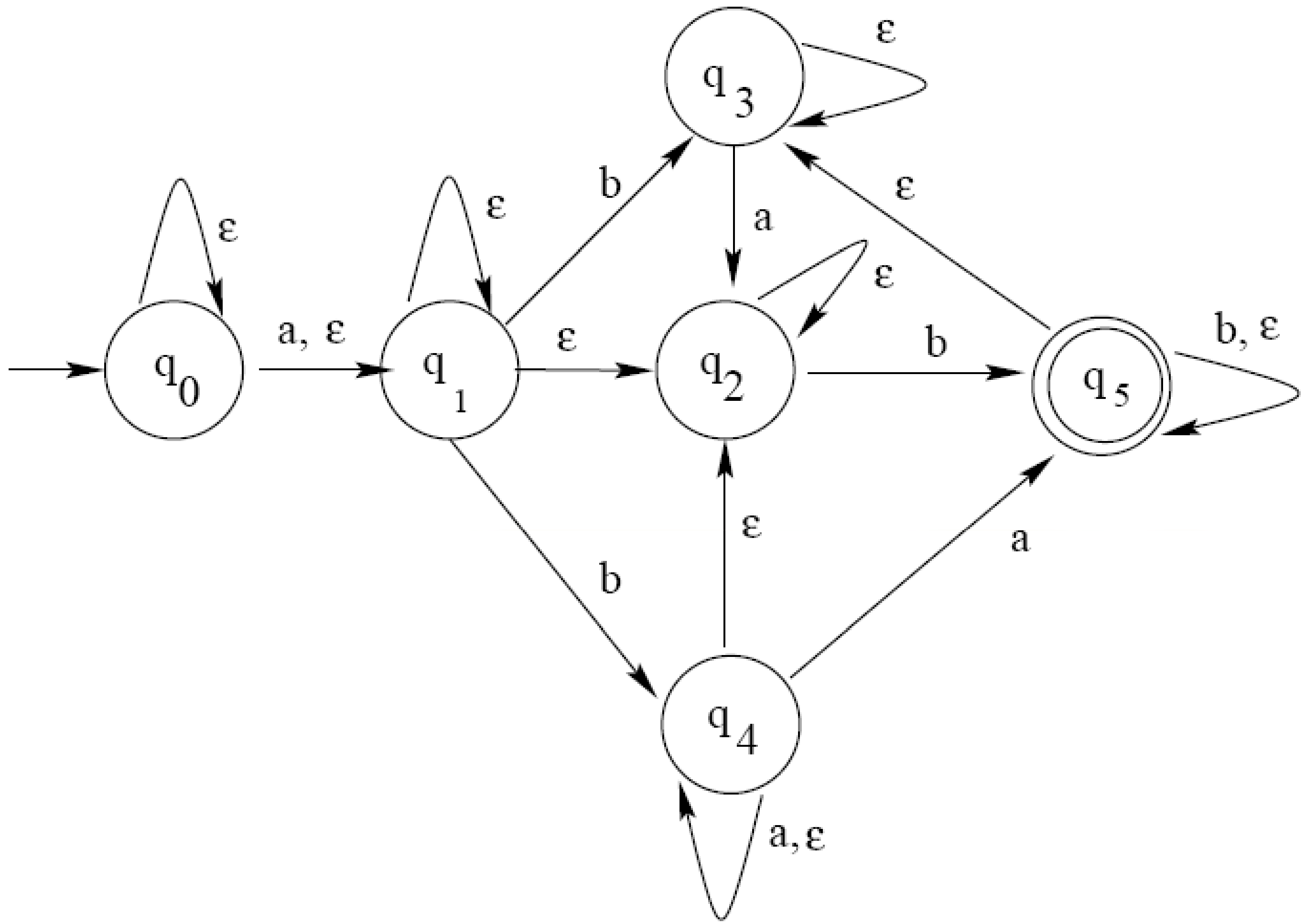


Autómata **no** minimizado

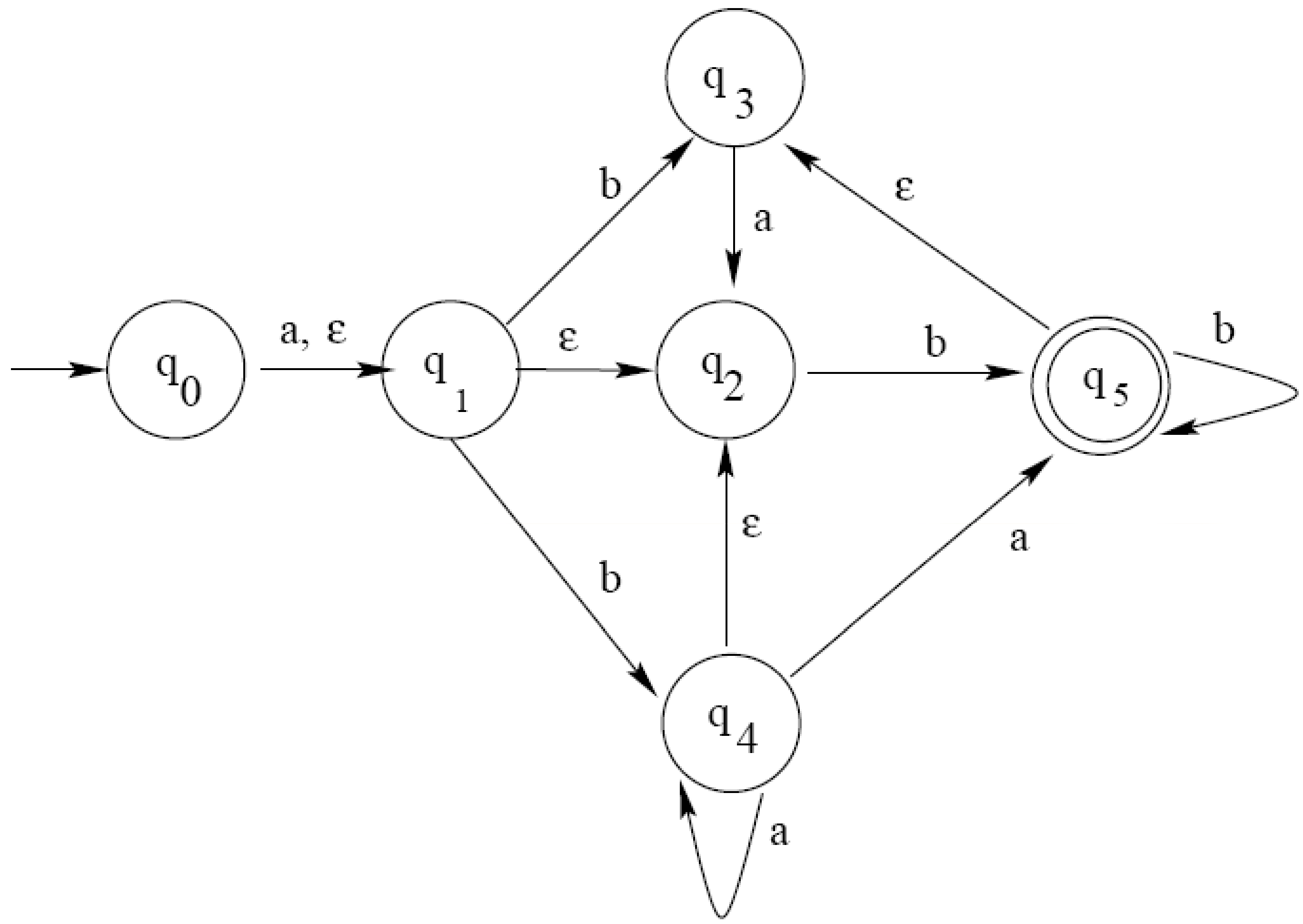
Minimización de un autómata



Autómata **minimizado**



Autómata finito **no** determinista
(**con** transiciones épsilon triviales)



Autómata finito **no** determinista
(**sin** transiciones épsilon triviales)

Algoritmo de Construcción de Subconjuntos

inicio

$p_0 \leftarrow clausura - \varepsilon(q_0)$

$Q_D \leftarrow \{p_0\}$ *y* p_0 *no marcado*

mientras *haya un estado* $p \in Q_D$ *<u>no</u> marcado* ***hacer***

Marcar a p

para *cada* $\sigma \in \Sigma$ ***hacer***

$p' \leftarrow clausura - \varepsilon(\delta_N(p, \sigma))$

si $p' \notin Q_D$

entonces

$Q_D \leftarrow Q_D \cup \{p'\}$ *y* p' *no marcado*

fin si

Definir $\delta_D(p, \sigma) \leftarrow p'$

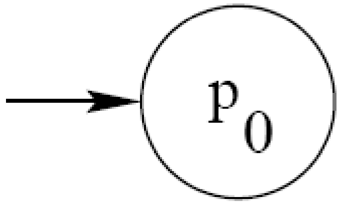
fin para

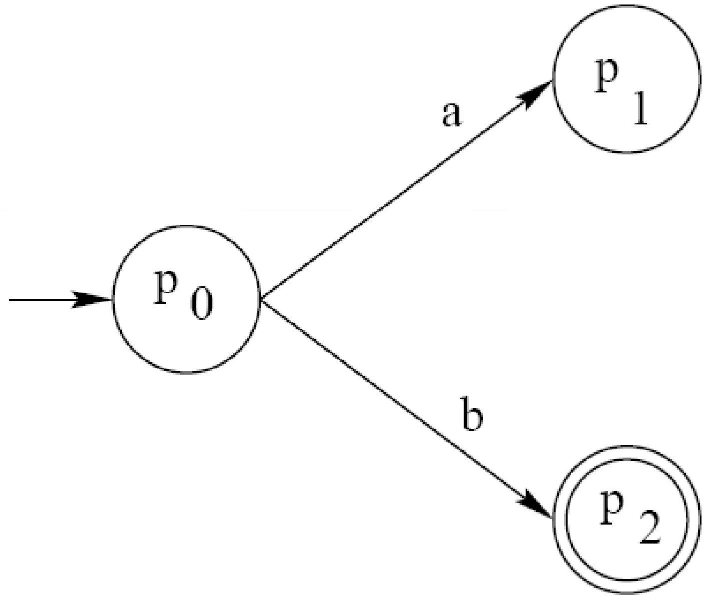
fin mientras

$F_D \leftarrow \{p_i \mid F_N \cap p_i \neq \varnothing\}$

fin

Algoritmo de Construcción de Subconjuntos: ejemplo

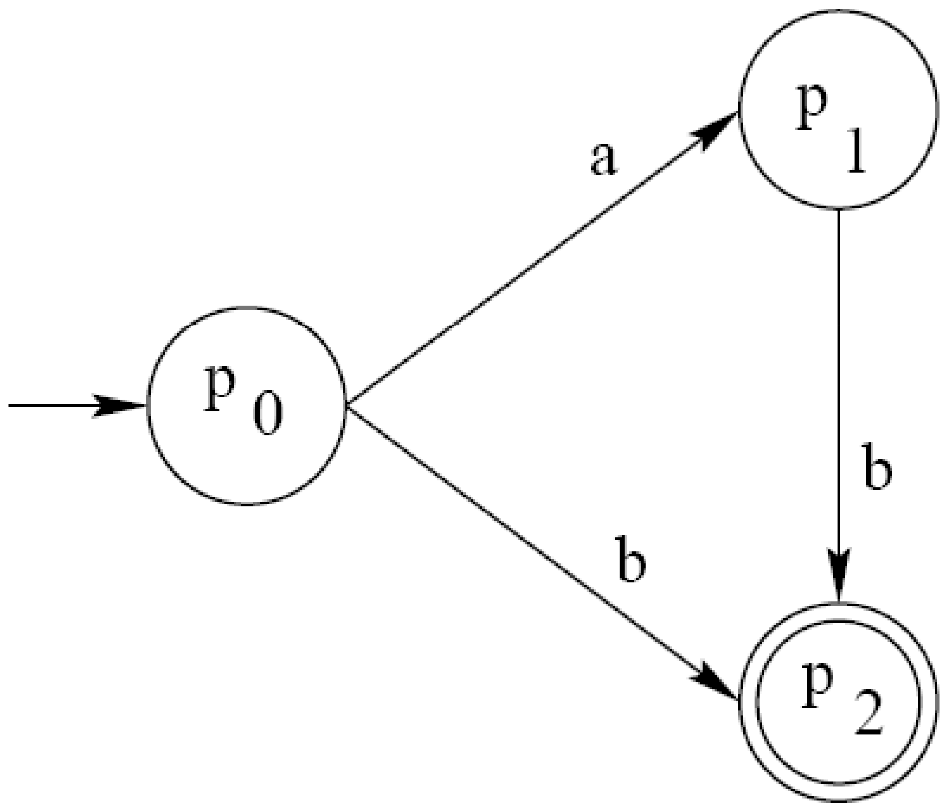




Estado inicial: p_0

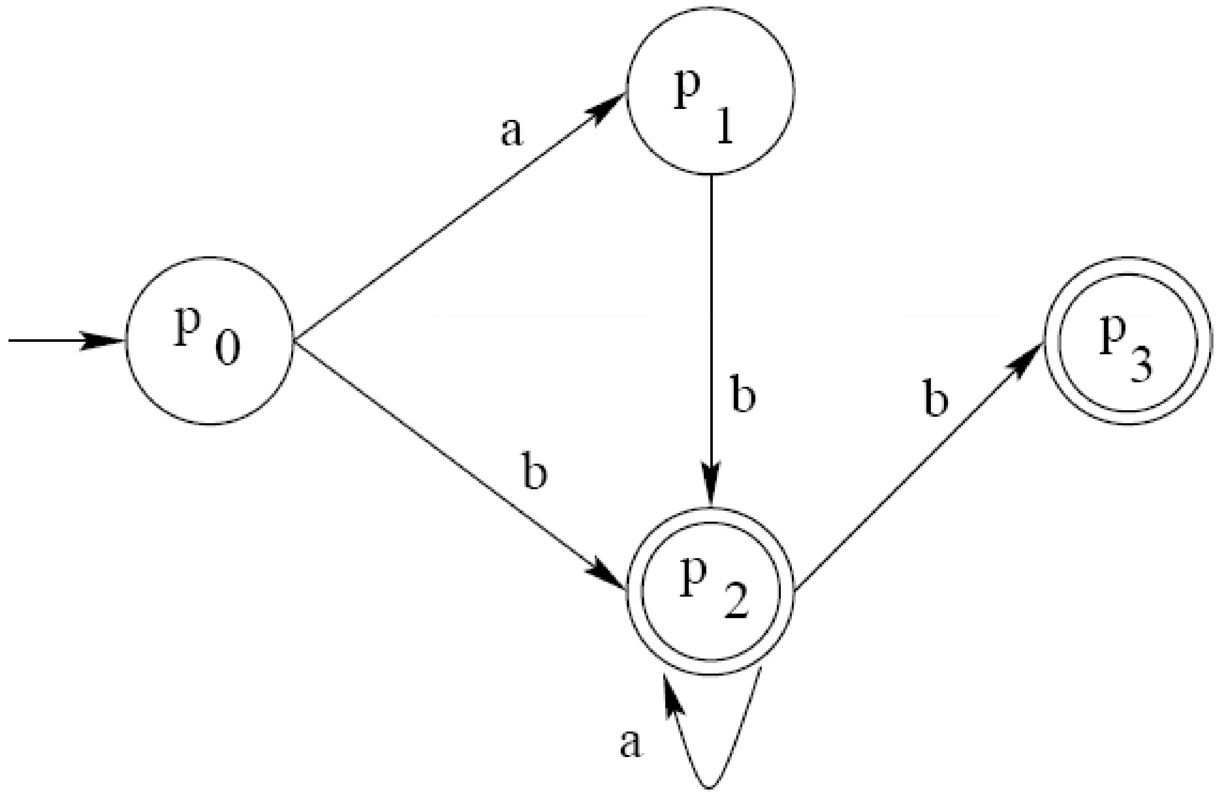
Transiciones de p_0

Algoritmo de Construcción de Subconjuntos: ejemplo



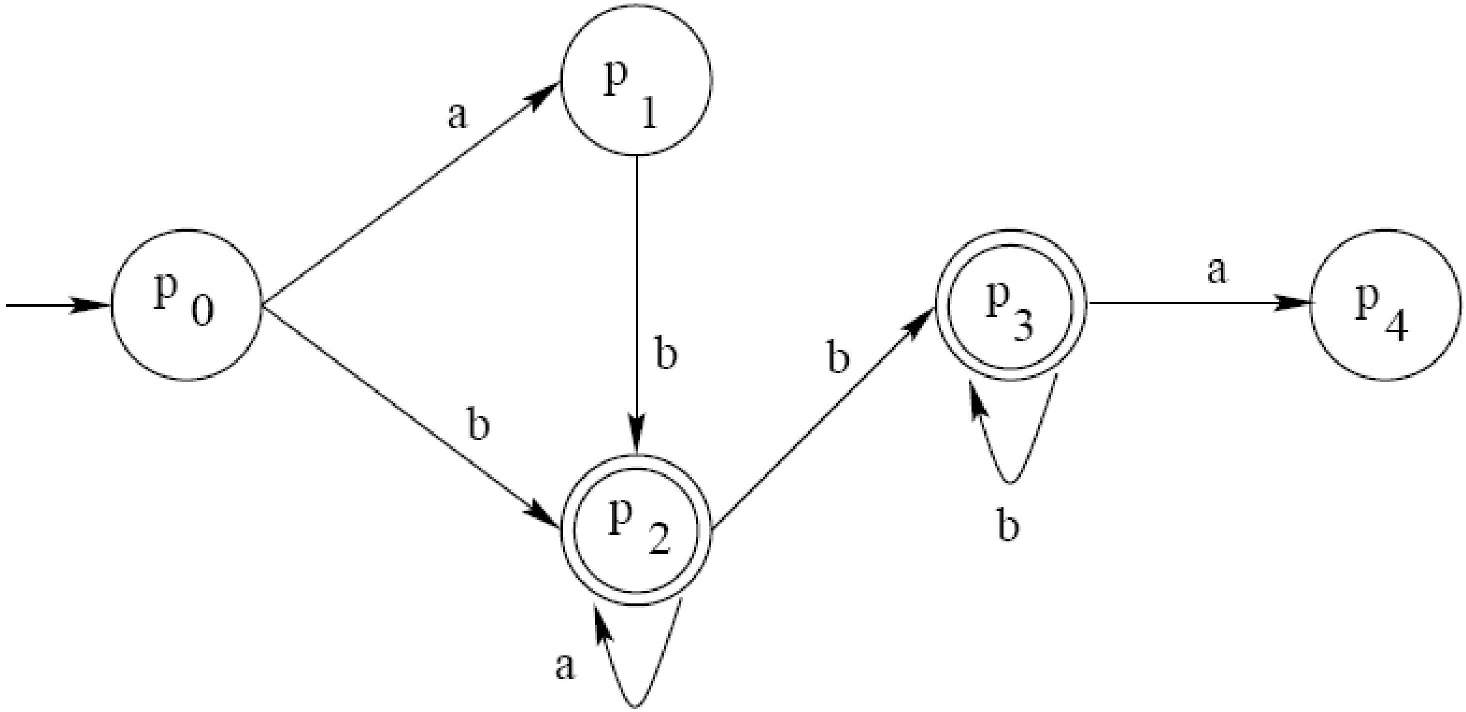
Transiciones de p_1

Algoritmo de Construcción de Subconjuntos: ejemplo



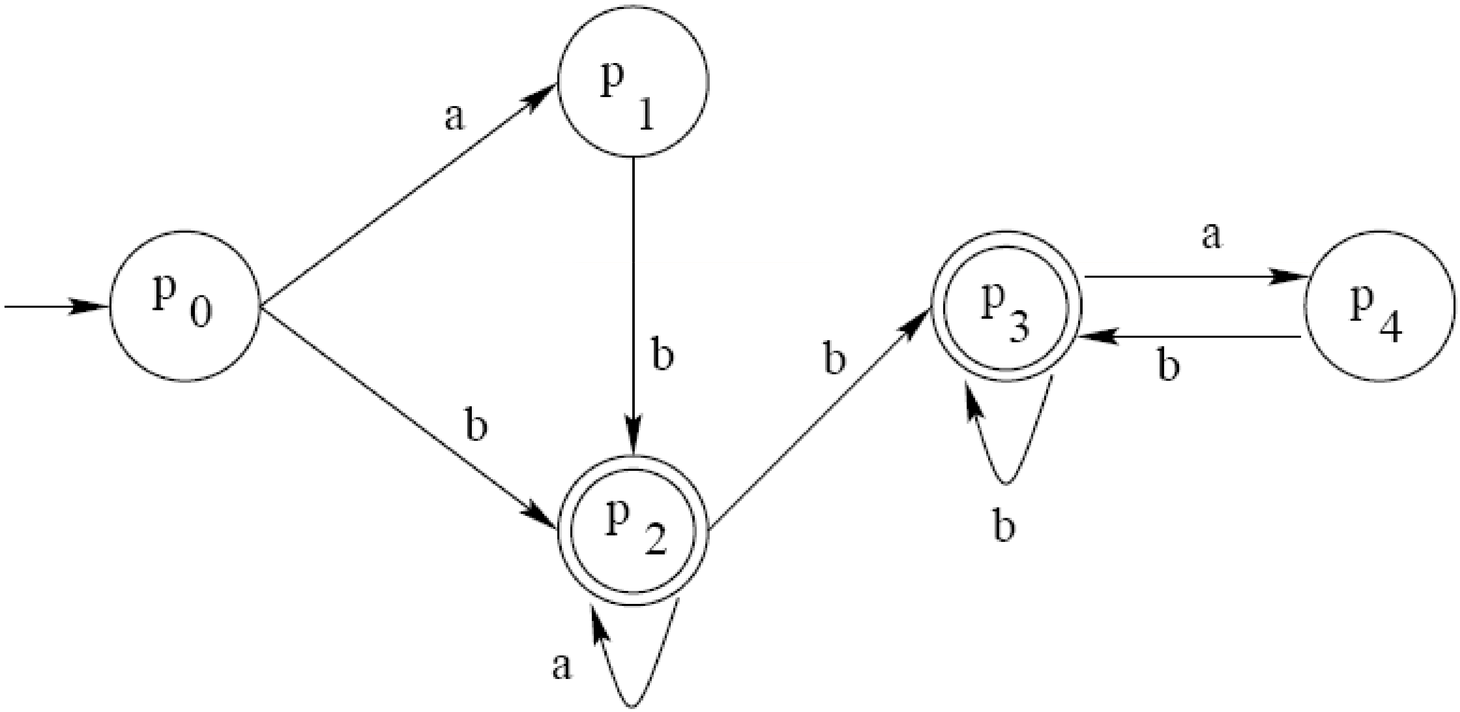
Transiciones de p_2

Algoritmo de Construcción de Subconjuntos: ejemplo



Transiciones de p $_3$

Algoritmo de Construcción de Subconjuntos: ejemplo



Transiciones de p 4

Ejercicio: transformación de una Gramática regular en un Autómata Finito Determinista

$P = \{$

$S \rightarrow \boldsymbol{a}\, A$

$A \rightarrow \boldsymbol{a}\, A$

$A \rightarrow \boldsymbol{a}$

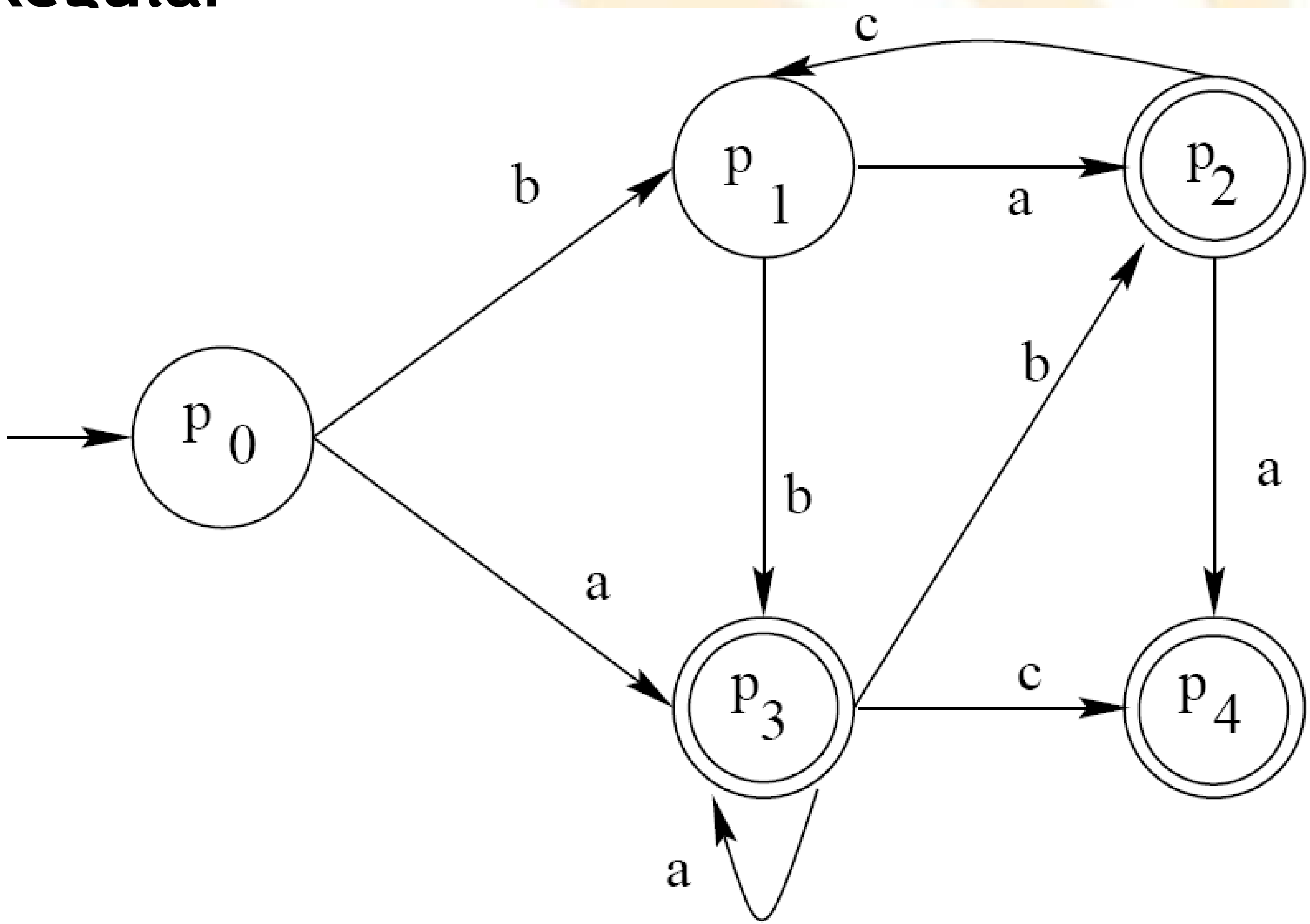
$A \rightarrow \boldsymbol{b}\, B$

$B \rightarrow \boldsymbol{b}\, B$

$B \rightarrow \boldsymbol{b}$

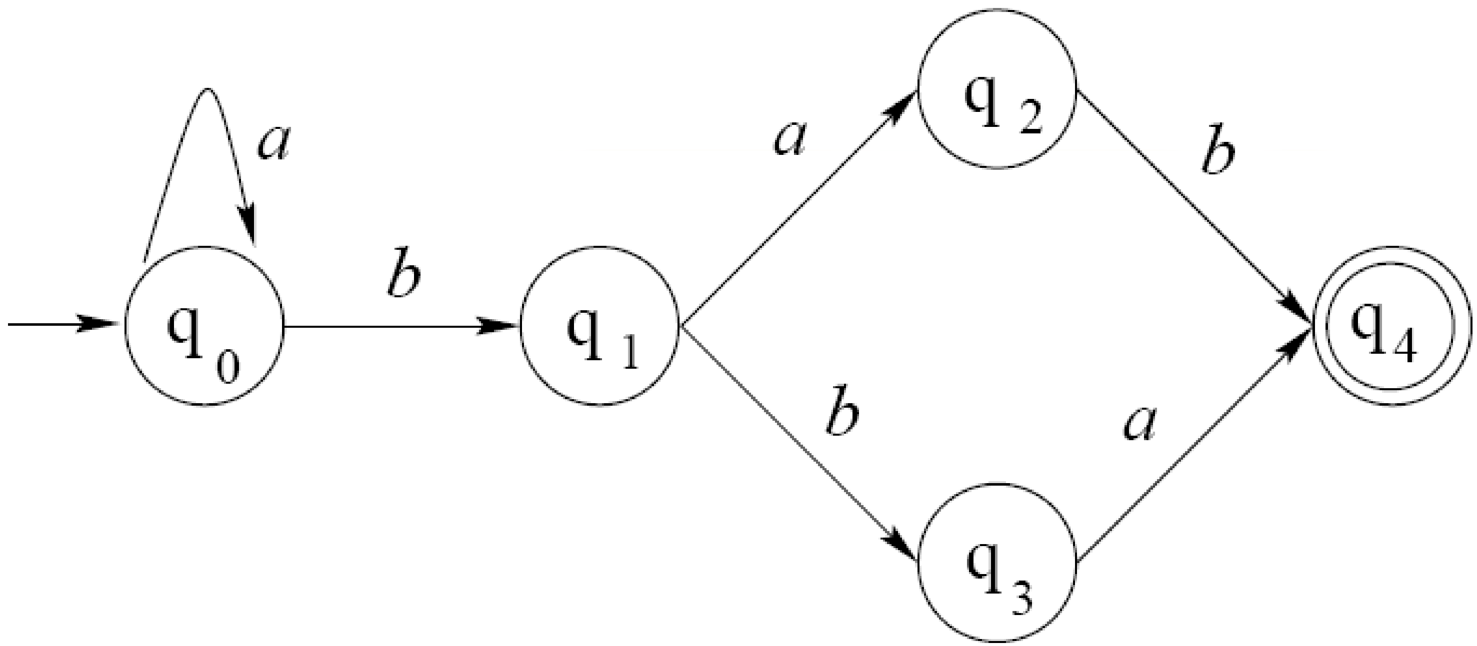
$\}$

Ejercicio: transformación de un Autómata Finito Determinista en una Gramática Regular



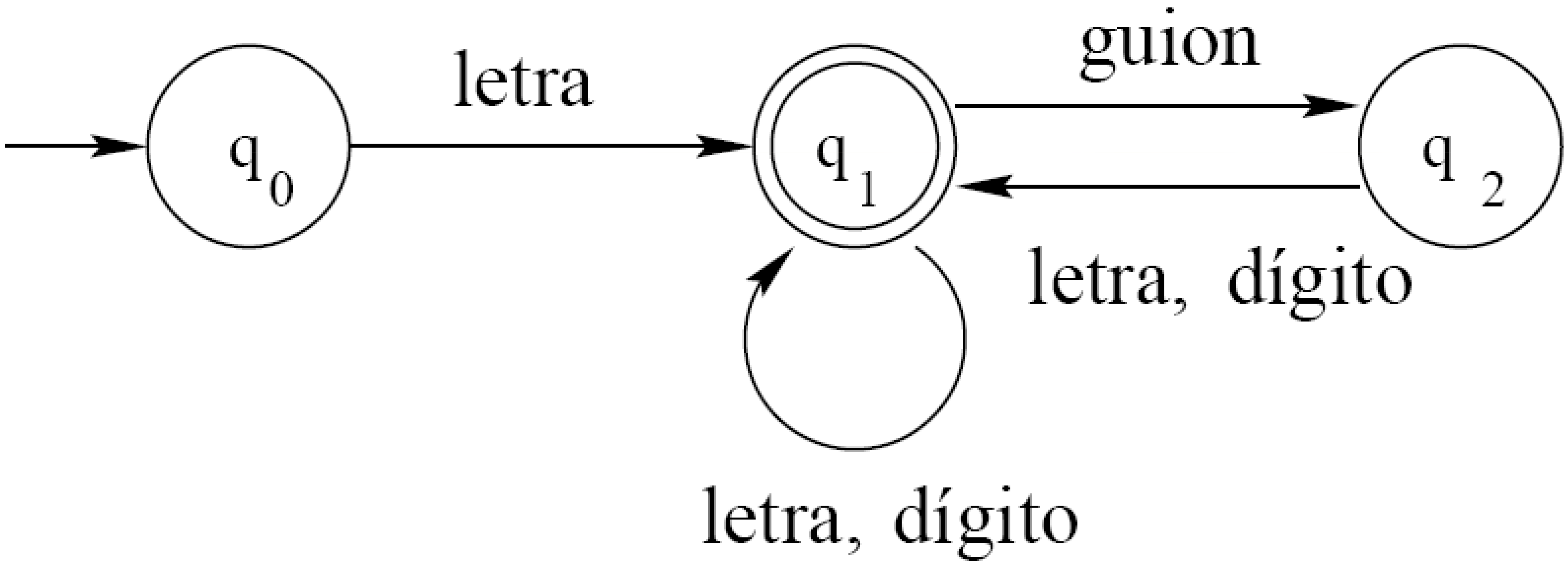
Problema de Análisis

Ejercicio: obtención de la Expresión Regular equivalente a un Autómata Finito Determinista



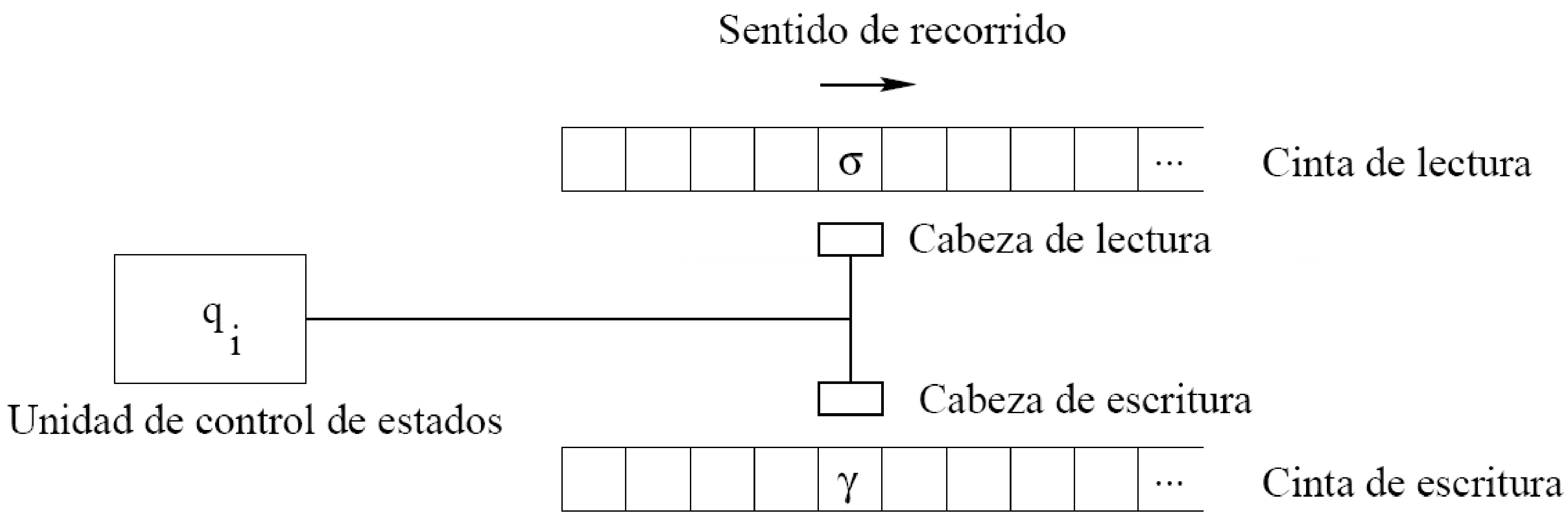
Problema de Análisis

Ejercicio: obtención de la Expresión Regular equivalente a un Autómata Finito Determinista

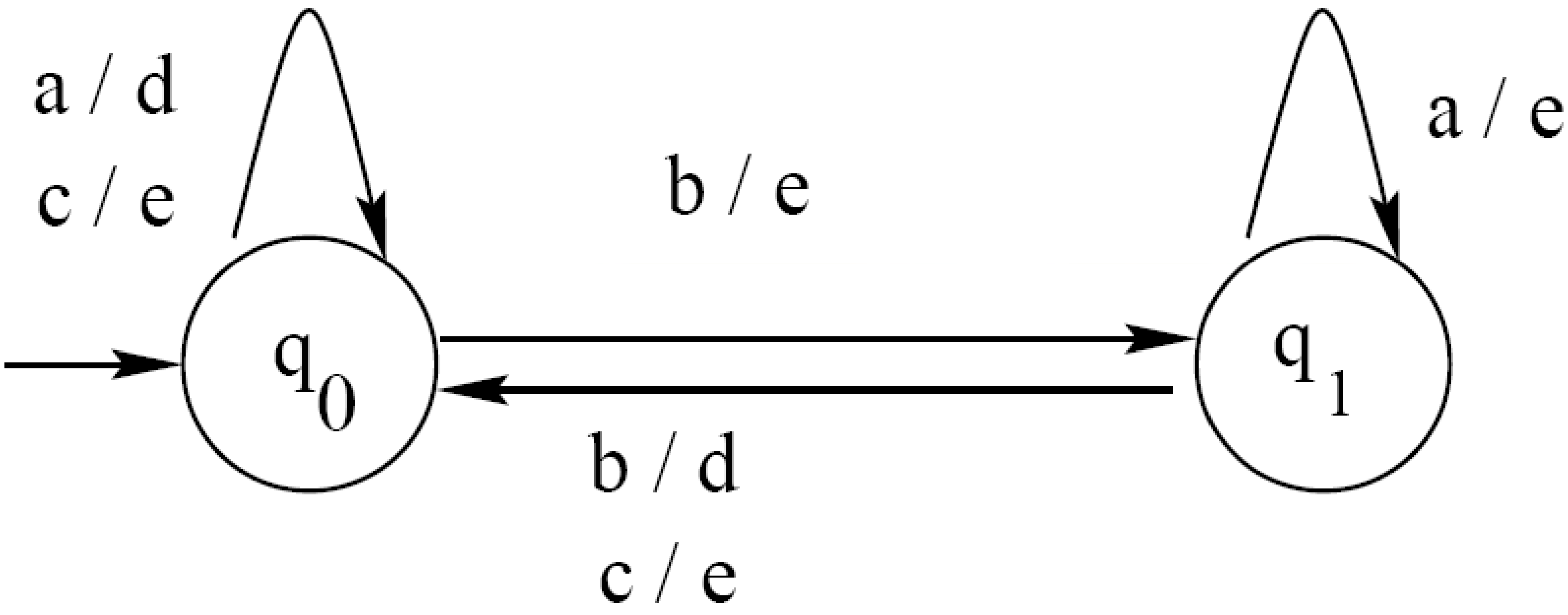


AFD que reconoce identificadores de COBOL

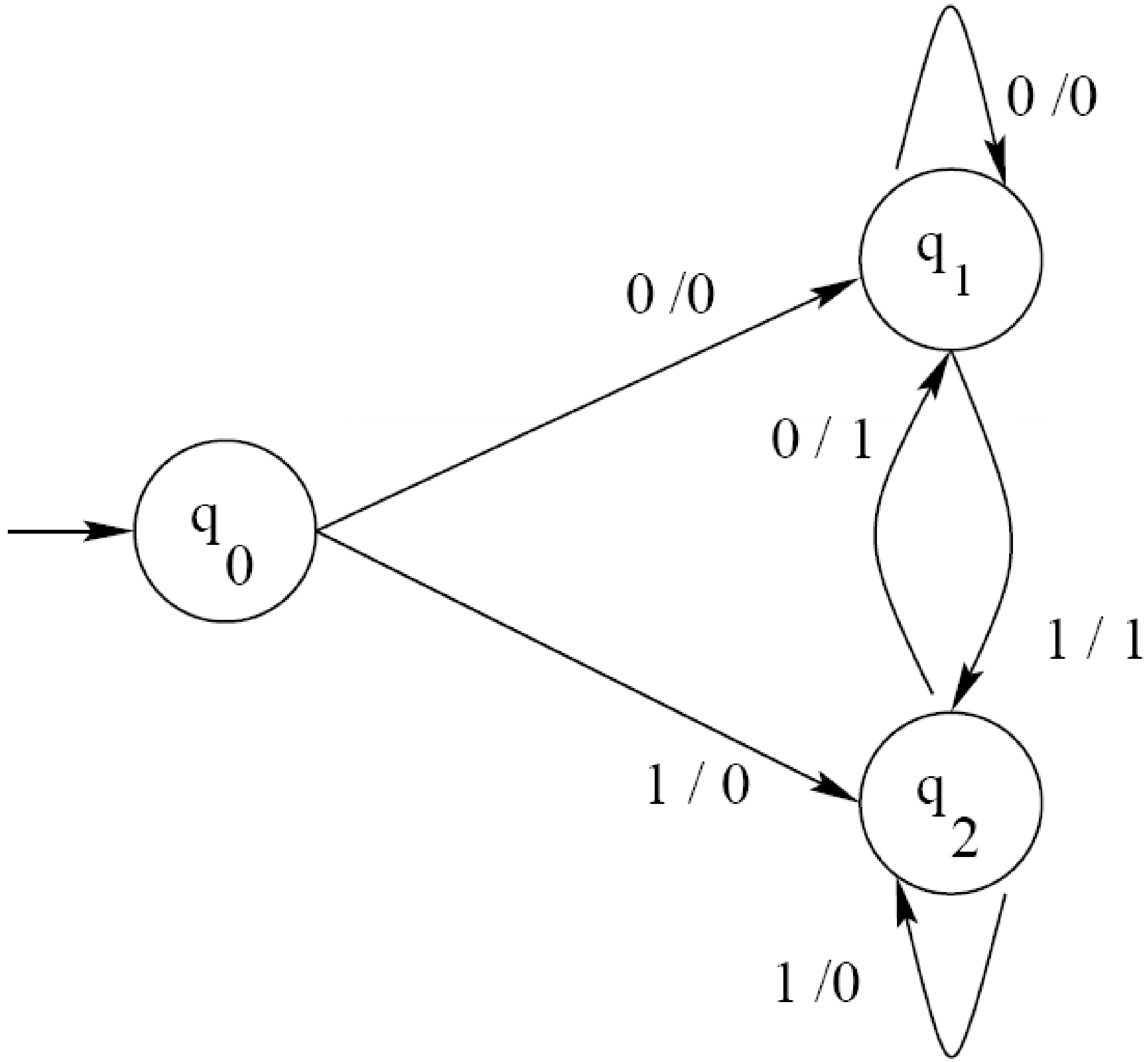
Componentes de una máquina secuencial



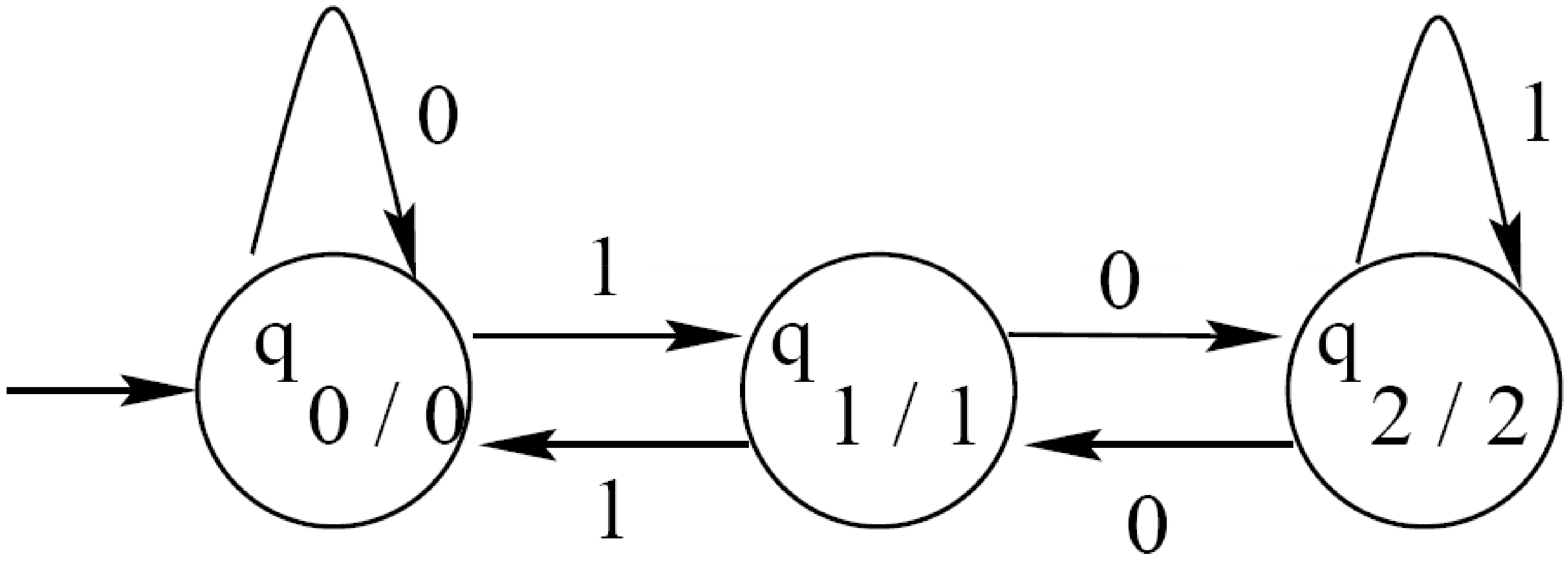
Representación gráfica de una máquina secuencial de Mealy



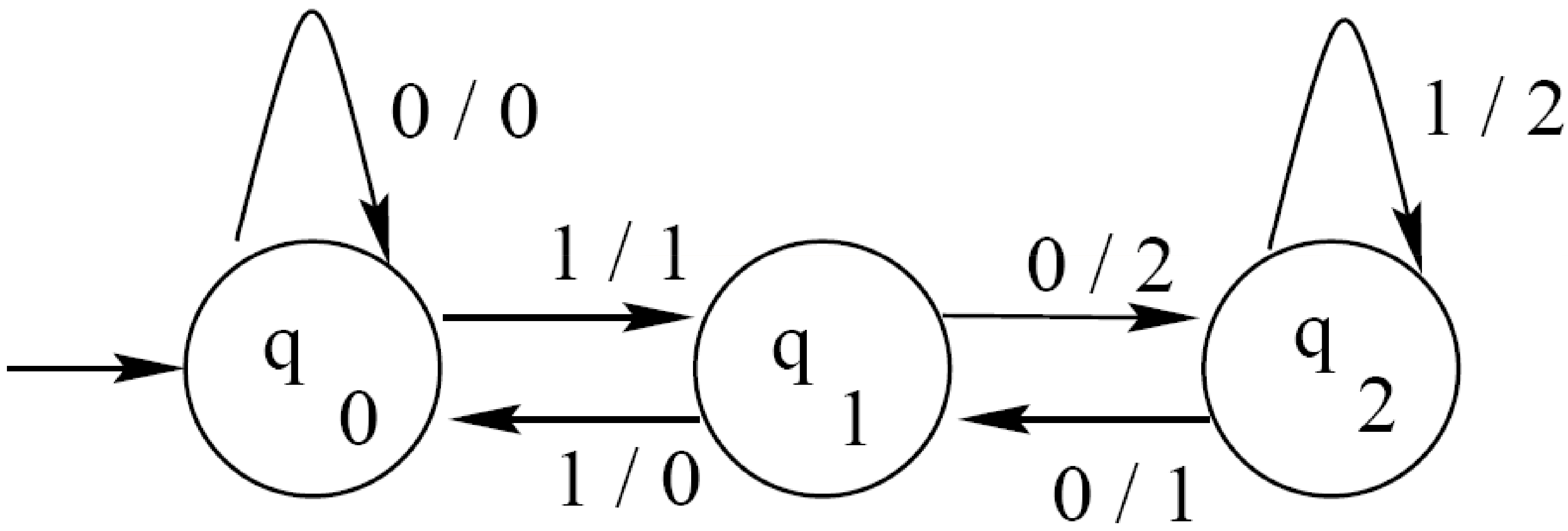
Máquina secuencial de Mealy que detecta



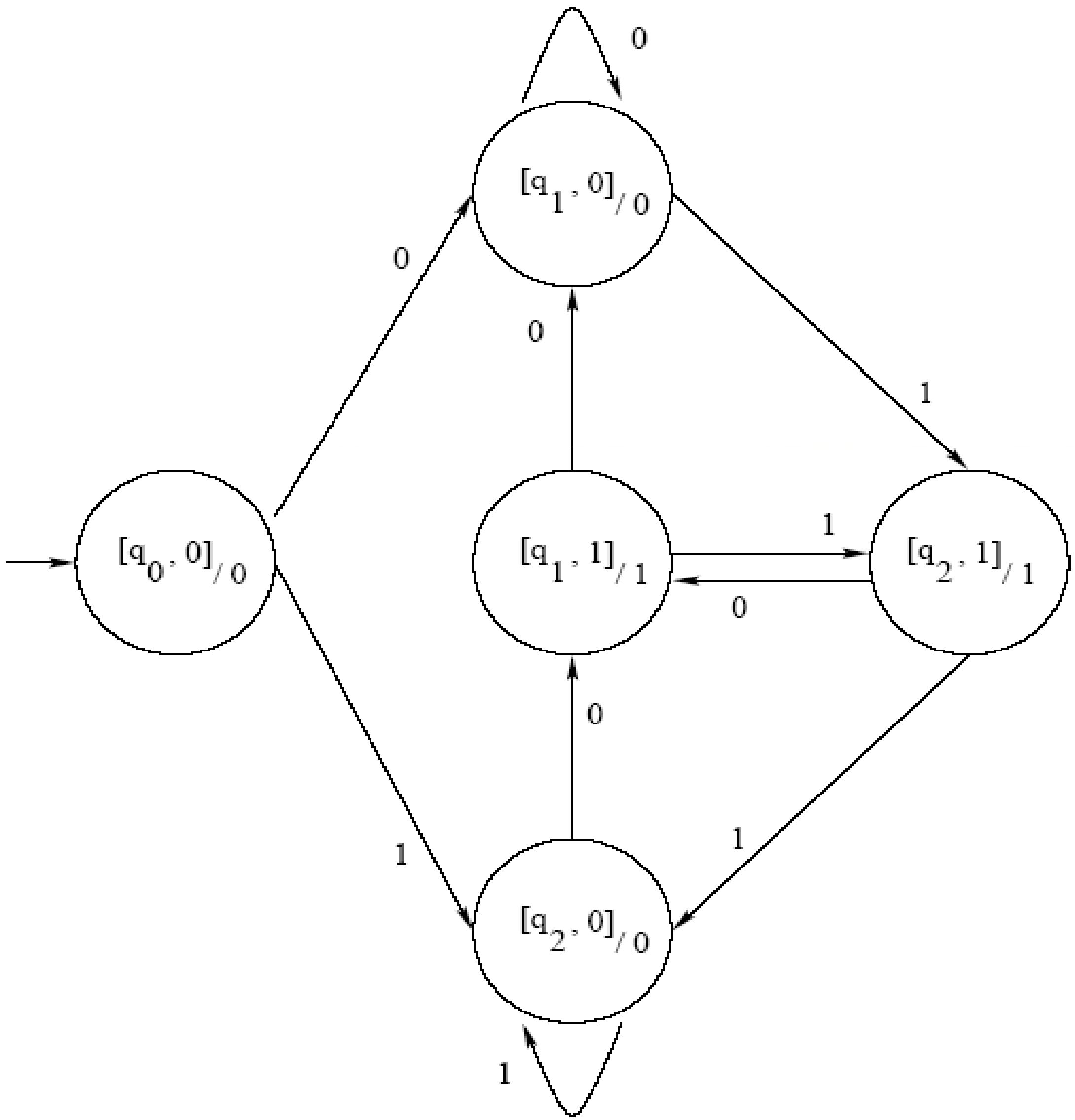
Máquina secuencial de Moore que calcula “n mod 3”



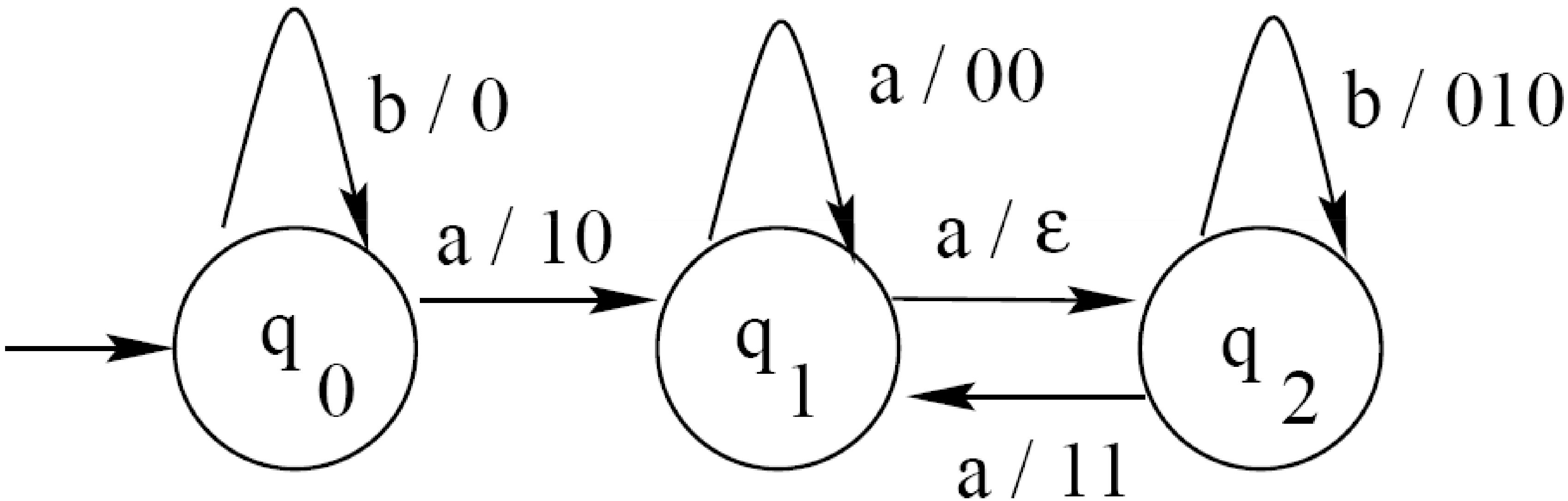
Máquina secuencial de Mealy que calcula "n mod 3"



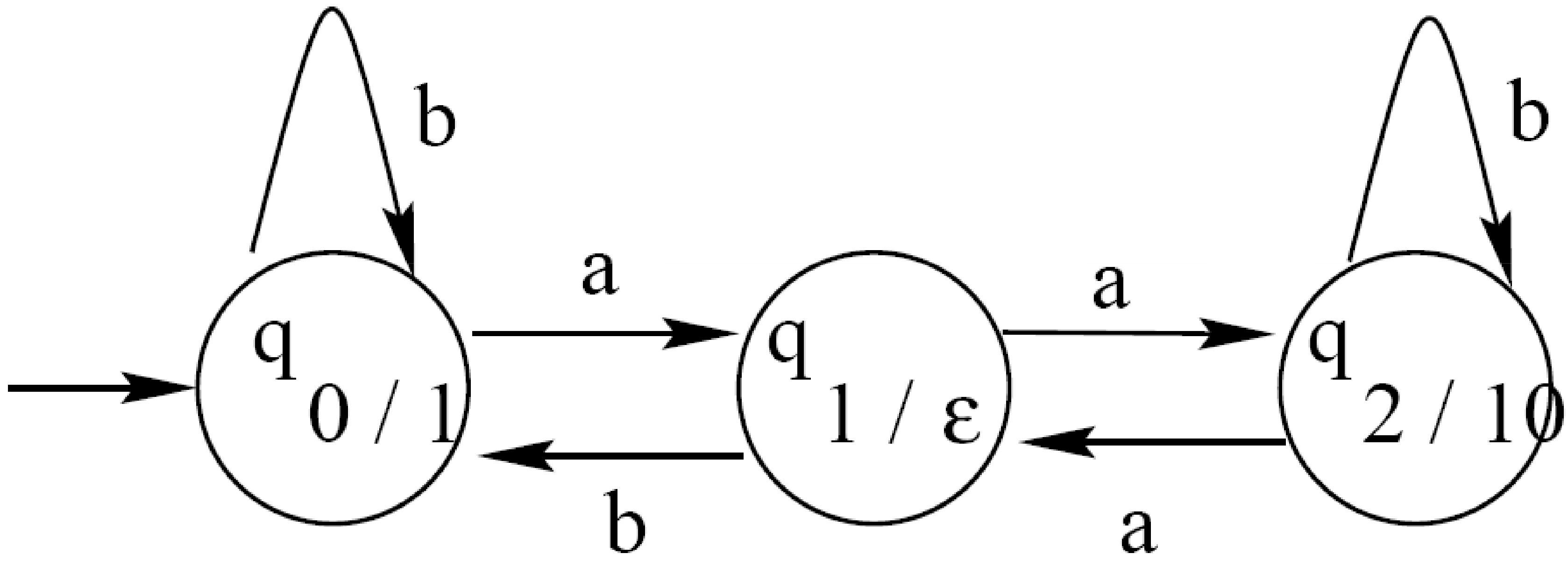
Máquina secuencial de Moore que detecta



Máquina secuencial de Mealy “generalizada”



Máquina secuencial de Moore "generalizada"



UNIVERSIDAD DE CÓRDOBA
ESCUELA POLITÉCNICA SUPERIOR
DEPARTAMENTO DE INFORMÁTICA Y ANÁLISIS NUMÉRICO

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

SEGUNDO CURSO, SEGUNDO CUATRIMESTRE

TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES

Tema 6.- Autómatas finitos y máquinas secuenciales