4.1 Máquinas de Turing

4. Lenguajes recursivos y recursivamente enumerables

Adaptación de transparencias de Fernando Rosa Velardo
(a su vez, traducción y adaptación de las de Ananth Kalyanaraman
(http://www.eecs.wsu.edu/~ananth/))

Máquinas de Turing

- Modelo equivalente a un ordenador
  - En cuanto a potencia de cómputo
    - O capacidad descriptiva
      - ¡No hablamos de eficiencia!

- Lo que no pueda computarse/resolverse con una MT no es computable/resoluble
  - Lo que no pueda decidir una MT no es **decidible**

Idea de Turing al definir las MTs

- Formalizar el comportamiento humano al usar lápiz y papel para resolver un problema
  - Nº finito de estados mentales (según Turing)
  - Papel cuadriculado: todo el necesario
    - Cinta infinita dividida en celdas que contienen símbolos
  - En un solo paso de cómputo podemos:
    1) Cambiar de estado (mental)
    2) Alterar el contenido de la cuadrícula que miramos
    3) Y mirar a otra casilla (la de la dcha. o la de la izda.)

Componentes de una MT (idea)

- $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ con $\Sigma \subset \Gamma$ y $B \in \Gamma$

B: blanco (símbolo especial)

Símbolos de entrada (y salida)

…	B	B	a_1	a_2	…	a_i	…	a_n	B	B	…

Cabeza lectora/escritora
(apunta a la **casilla de trabajo**;
a lo que miramos)

q

Estado

Componentes de una MT

- $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$
  - Q es el conjunto finito de estados
  - Σ es el alfabeto de entrada
  - Γ es el alfabeto de cinta ($\Sigma \subset \Gamma$)
  - $q_0 \in Q$ es el estado inicial
  - $B \in \Gamma \backslash \Sigma$ es el símbolo del blanco
  - $F \subseteq Q$ es el conjunto de estados de aceptación
  - δ: Q x Γ --- ---> Q x Γ x {L, R} es la fc. de transición

Función de transición de una MT

- Un paso de cómputo hace lo siguiente:
  - Si **q** es el estado actual

  - **X** es el símbolo contenido en la casilla de trabajo
  - Y la función de transición dice **δ(q, X) = (p, Y, D)**
    - Cambiamos el estado actual de **q** a **p**
    - Sustituimos el contenido **X** de la casilla de trabajo por **Y**
    - Movemos la cabeza lectora/escritora en la dirección **D**
      - Si D = "L" o D = "←" una posición a la izquierda *(left)*
      - Si D = "R" o D = "→" una posición a la derecha *(right)*

Configuraciones en una MT

- Una configuración (o descripción instantánea) describe a la MT en un momento dado
  - $a_1a_2...a_{i-1}qa_ia_{i+1}...a_n$ o $(a_1a_2...a_{i-1}, q, a_ia_{i+1}...a_n)$
    - q es el estado actual
    - a_i es el símbolo contenido en la casilla de trabajo
    - $a_1a_2...a_{i-1}a_ia_{i+1}...a_n$ es el contenido de la cinta
  - $a_1a_2...a_{i-1}qa_ia_{i+1}...a_n$ |--- (o $\rightarrow$) (paso de cómputo)
    - $a_1a_2...a_{i-1}Y\ p\ a_{i+1}...a_n$ si $\delta(q, a_i) = (p, Y, R)$
    - $a_1a_2...\ p\ a_{i-1}Y\ a_{i+1}...a_n$ si $\delta(q, a_i) = (p, Y, L)$

Lenguaje reconocido por una MT

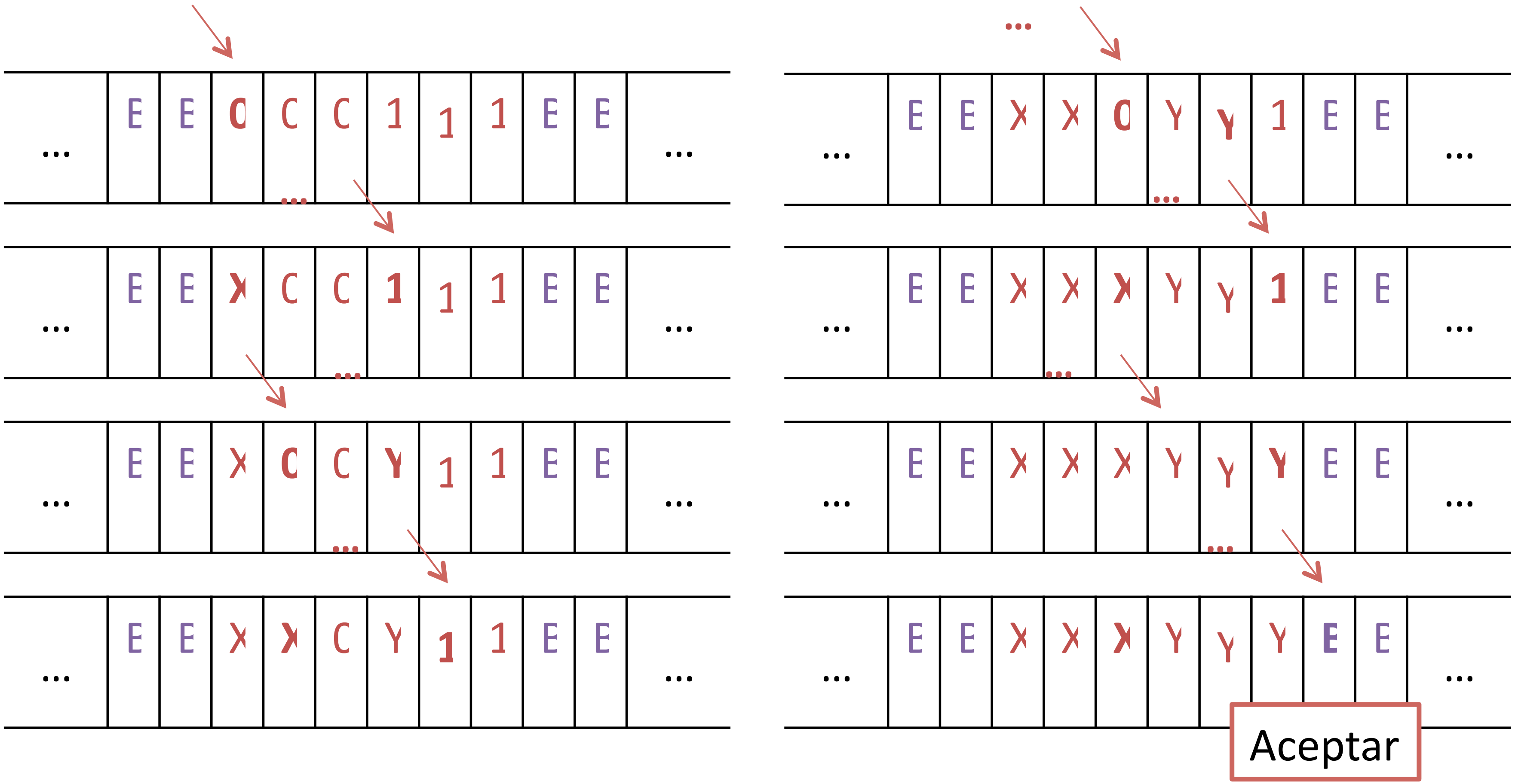
- ¿Pertenece *x* al lenguaje descrito por una MT?
  - Se coloca la cadena de entrada *x* en la cinta, precedida y seguida por infinitos blancos, con la cabeza apuntando al primer símbolo de *x*
  - Si la MT **para**:

    para porque llega a una configuración en la que la función de transición no está definida

    - **acepta** *x* si lo hace en un estado final (o de aceptación)
    - **rechaza** *x* si lo hace en un estado no final
  - Las MTs pueden no parar…
    - En ese caso diremos que **cicla** (con la entrada *x*)

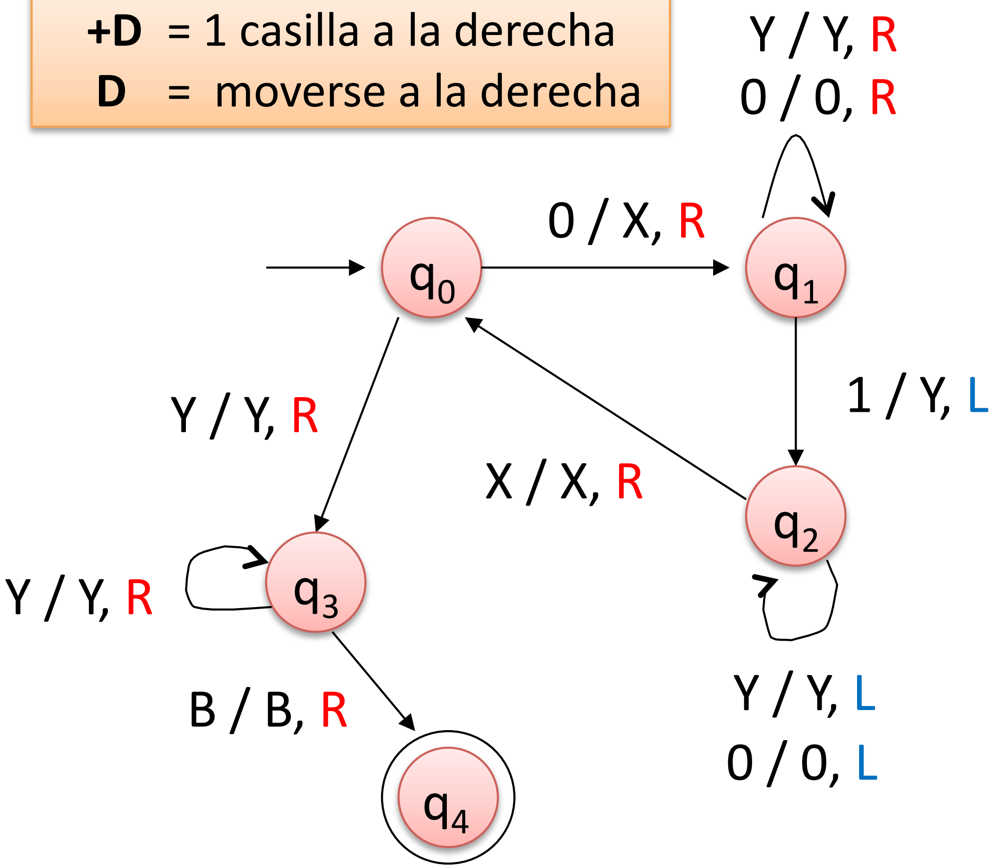
Ejemplo 1: $L = \{0^n1^n \mid n \geq 1\}$

- Idea para x = 000111



Ejemplo 1: $L = \{0^n1^n \mid n \geq 1\}$

+D = 1 casilla a la derecha
D = moverse a la derecha



1. Marcamos con X el siguiente 0 **+D**
2. **D** hasta el 1º 1 y lo marcamos con Y **+I**
3. **I** hasta encontrar X **+D**
4. Si leemos un 0 pasamos a 1. Si no, **D** para comprobar que no hay más 1's. Si es así, nos movemos hasta el siguiente B, paramos y aceptamos

Probar con ejemplos: ε, 01, 001, 011, 10, etc.

Ejemplo 1: $L = \{0^n 1^n \mid n \geq 1\}$

- Representación tabular de la fc. de transición

	Siguiente símbolo de cinta				
Estado actual	**0**	**1**	**X**	**Y**	**B**
> q_0	(q_1, X, R)	-	-	(q_3, Y, R)	-
q_1	$(q_1, 0, R)$	(q_2, Y, L)	-	(q_1, Y, R)	-
q_2	$(q_2, 0, L)$	-	(q_0, X, R)	(q_2, Y, L)	-
q_3	-	-	-	(q_3, Y, R)	(q_4, B, R)
$*q_4$	-	-	-	-	-

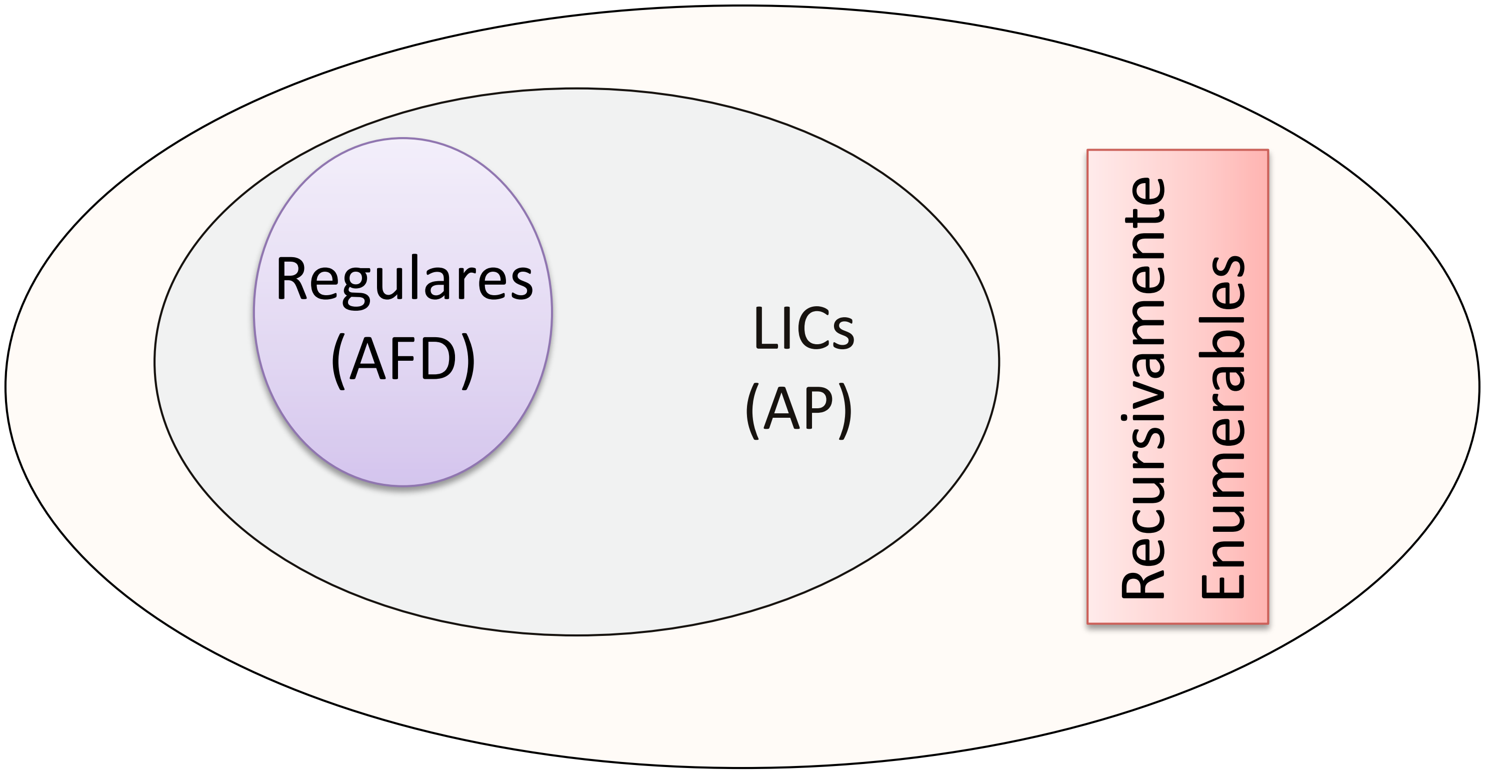
Más vueltas

- Probad siempre con cadenas sencillitas, a ver si se reconocen o se rechazan
- ¿Qué hacer si quisiésemos que también se reconociese la cadena vacía?
- ¿Cómo modificar la idea para reconocer el lenguaje no independiente del contexto

$$\{a^n b^n c^n \mid n \geq 1\} \text{ o } \{a^n b^n c^n \mid n \geq 0\}?$$

Lenguajes reconocidos por MTs

- **Lenguajes recursivamente enumerables (RE)**



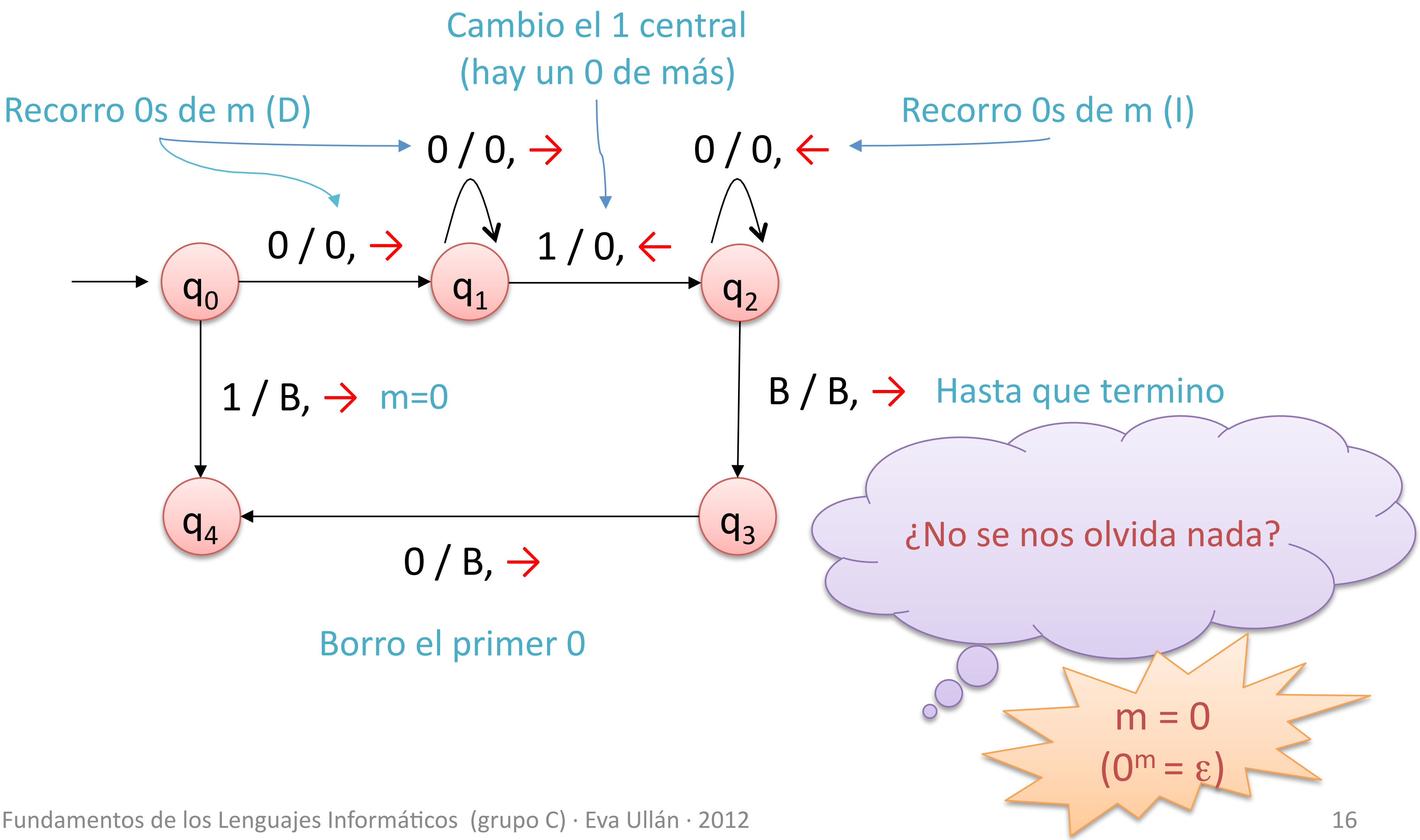
MTs para cálculos

- Aunque nuestro propósito fundamental es el de describir lenguajes
  - Las MTs nos sirven para ello
    - Lenguaje reconocido = conjunto de cadenas aceptadas
- El propósito original de Turing era el de calcular/computar funciones sobre naturales
  - Los naturales se representaban como bloques de un solo carácter (código unario)
    - Por ejemplo, n podría representarse como 0^n

Ejemplo 2: suma

- Función binaria: hay que separar argumentos
  - entrada: $0^m 1 0^n$ salida: 0^{m+n}
  - ¿Ideas?
    - Cambiamos el 1 central por un 0
    - Y borramos un 0

Ejemplo 2: m+n

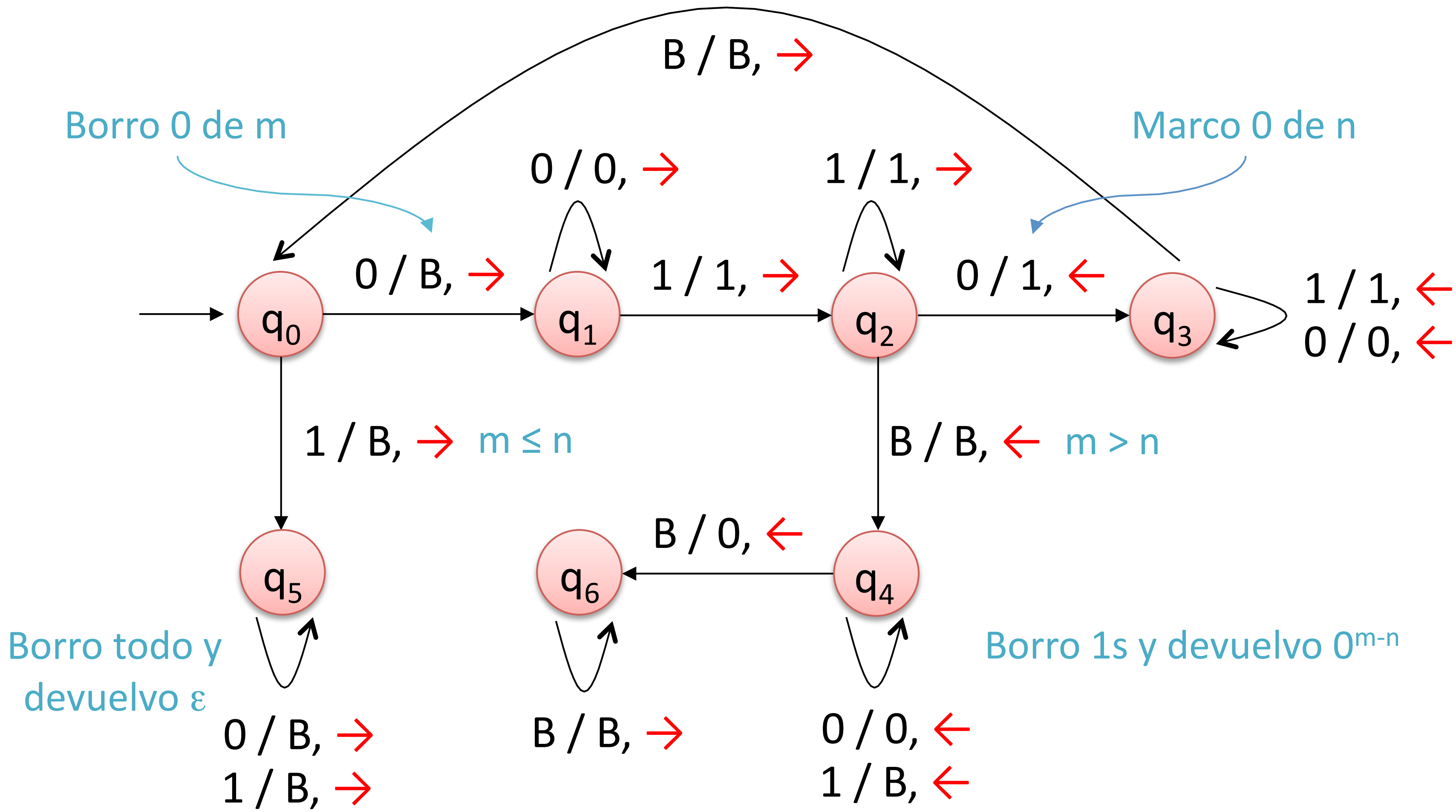

Cambio el 1 central
(hay un 0 de más)
Recorro 0s de m (D)
Recorro 0s de m (I)
0 / 0, →
0 / 0, ←
0 / 0, →
1 / 0, ←
q0
q1
q2
1 / B, → m=0
B / B, → Hasta que termino
q4
q3
0 / B, →
Borro el primer 0
¿No se nos olvida nada?
m = 0
(0m = ε)

Ejemplo 3: resta *propia*

- "m -- n" = max{m-n, 0}
  - entrada: 0^m10^n salida: 0^{m-n} o ...BB...B...

1. Por cada 0 de la izquierda (marcamos B –borramos-), buscamos el primer 0 de la derecha (y marcamos 1)
2. Repetir hasta que:
   1. No quedan 0s a la izquierda del 1 ($m \leq n$)
      El resultado es **0**: borrar también 0s sobrantes a la derecha del 1 (y el 1) y parar (devolvemos $0^0 = \varepsilon$ -ningún símbolo-)
   2. No quedan 0s a la derecha del 1 ($m > n$)
      El resultado es **m-n**: borrar los 1s y escribir un 0 de más (por el borrado de más) y parar (devolvemos 0^{m-n})

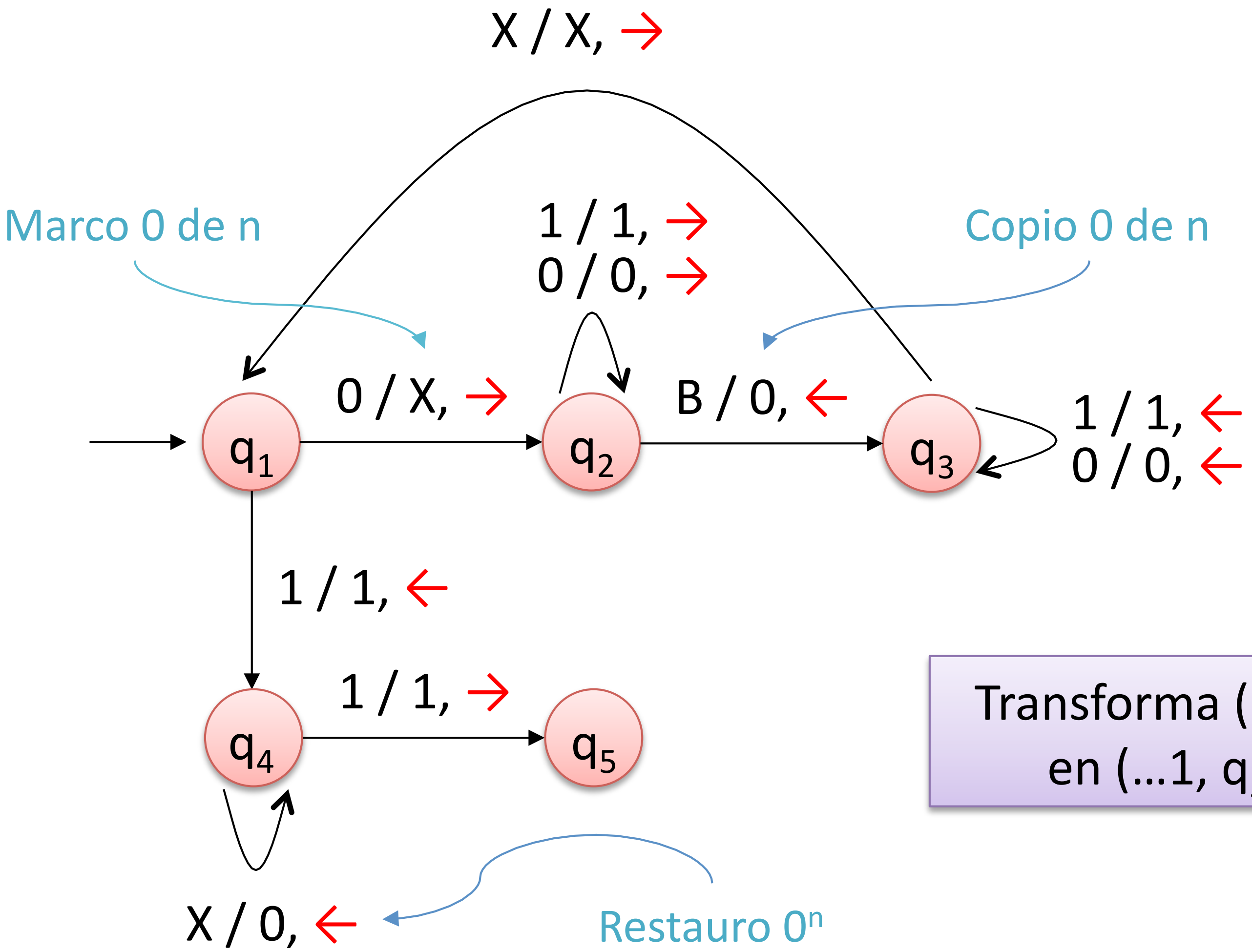
Ejemplo 3: “m -- n” = max{m-n, 0}



Ejemplo 4: Producto

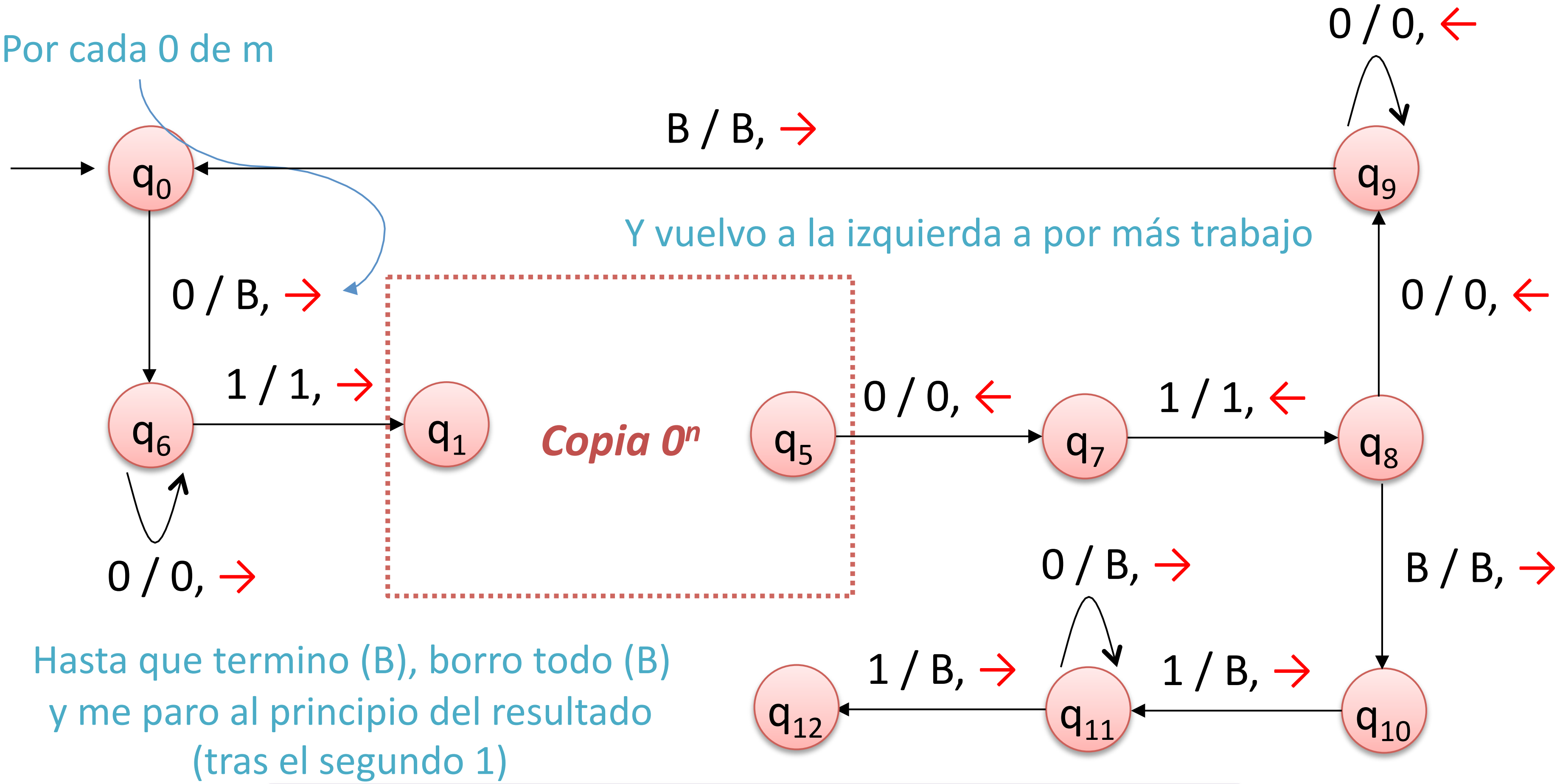
- 0^m10^n1 (entrada) 0^{mn} (salida)
  - » El 1 final puede escribirse antes (MT previa, p.ej.)
  - » Suponemos n,m > 0
- Idea:
  1. Por cada 0 de 0^m se escriben n 0s a la derecha del último 1 *(subrutina copia)*
  2. Una vez hecho lo anterior, se borra el 0 considerado (se sobreescribe con B)
  3. Tras completar lo anterior para cada 0, borramos los n 0s y los 1s

Ejemplo 4: Subrutina *copia*



Transforma (…1, q_1, 0^n1)
en (…1, q_5, 0^n10^n)

Ejemplo 4: Multiplicación



Transforma $(\varepsilon, q_0, 0^m10^n1)$ en $(\varepsilon, q_{12}, 0^{mn})$
(falta caso m=0; visto en clase)