# Máquinas de Turing

Senén Barro Ameneiro, CiTIUS

@SenenBarro

Material elaborado fundamentalmente por
el profesor Manuel Mucientes Molina

# Bibliografía


- J.E. Hopcroft, R. Motwani y J.D. Ullman, "Teoría de Autómatas, Lenguajes y Computación", Addison Wesley, 2008.
  - Capítulo 8

- P. Linz, ''An Introduction to Formal Languages and Automata'', Jones and Bartlett Publishers, Inc., 2001.
  - Capítulo 10

# Alan Turing

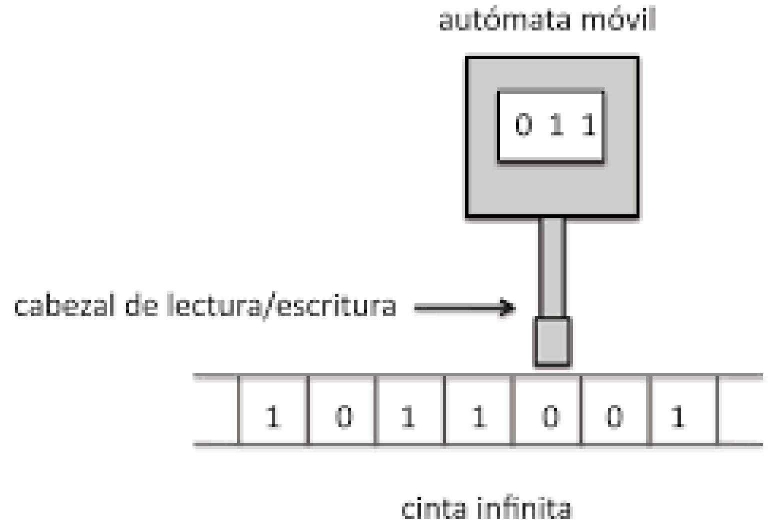
## "Máquinas inteligentes", 1948*

.. una ilimitada capacidad de memoria obtenida en la forma de una cinta infinita marcada con cuadrados, en cada uno de los cuales podría imprimirse un símbolo. En cualquier momento hay un símbolo en la máquina; llamado el símbolo leído. La máquina puede alterar el símbolo leído y su comportamiento está en parte determinado por ese símbolo, pero los símbolos en otros lugares de la cinta no afectan el comportamiento de la máquina. Sin embargo, la cinta se puede mover hacia adelante y hacia atrás a través de la máquina, siendo esto una de las operaciones elementales de la máquina. **Por lo tanto cualquier símbolo en la cinta puede tener finalmente una oportunidad.**

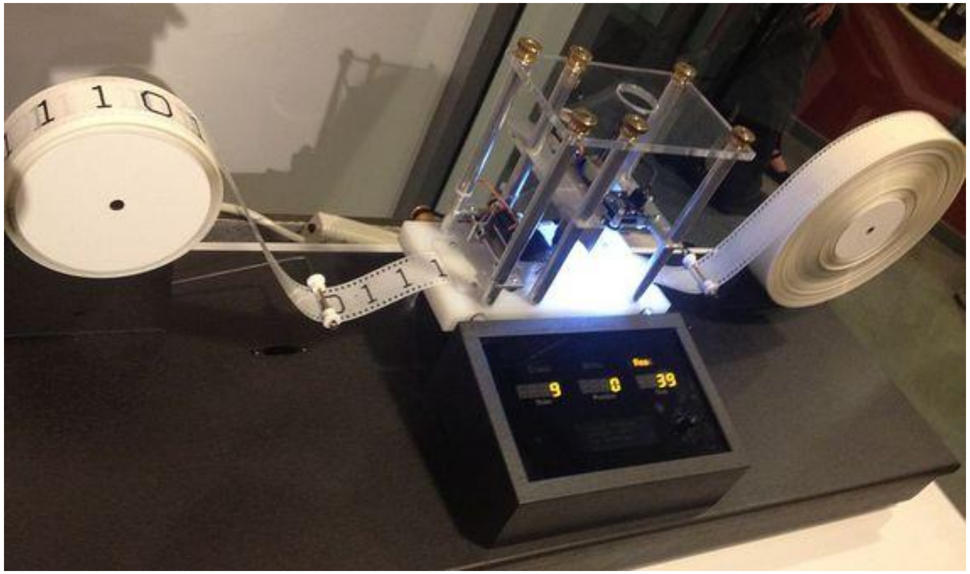
* Explicación de la "máquina de computación lógica", después conocida como: Máquina de Turing

# Máquina de Turing

Una Máquina de Turing (MT) es un autómata que cuenta con un dispositivo de almacenamiento denominado cinta.




Asociada con la cinta, existe una cabeza de lectura/escritura



# Máquina de Turing

- En una MT:
  - La entrada está escrita en la cinta al comienzo
  - La salida se escribirá en la cinta durante la operación de la MT

- $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$
  - $Q$: conjunto de estados
  - $\Sigma$: alfabeto de entrada
  - $\Gamma$: alfabeto de la cinta
  - $\delta$: función de transición
  - $q_0 \in Q$ : estado inicial
  - $B \in \Gamma$ : espacio en blanco ($B \notin \Sigma$)
  - $F \subseteq Q$: conjunto de estados finales

- $\Sigma \subseteq \Gamma - \{B\}$: el alfabeto de entrada es un subconjunto del alfabeto de cinta sin el espacio en blanco

# Máquina de Turing

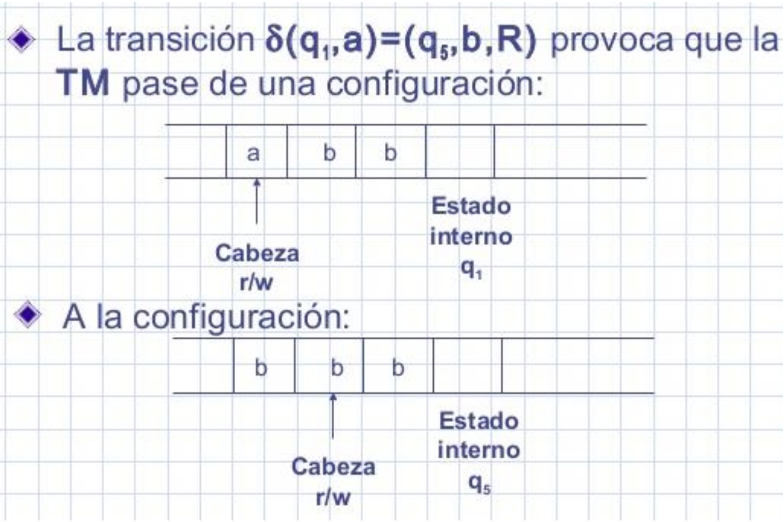
$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{I, D\}$

- Acciones:
  - Escribir el nuevo símbolo
  - Cambiar de estado
  - Mover la cabeza de lectura/escritura

- Movimiento tras la operación de lectura y escritura
- La cabeza se desplaza una única posición Izda/Dcha

Ejemplo:

$$\delta(q_1, a) = (q_5, b, R)$$



# Máquina de Turing: ejemplo

Ejemplo:

| | 0 | 1 | $\flat$ |
|---|---|---|---|
| $q_0$ | ($q_0$,0,R) | ($q_0$,1,R) | ($q_1$,$\flat$,L) |
| $q_1$ | – | ($q_2$,1,R) | – |

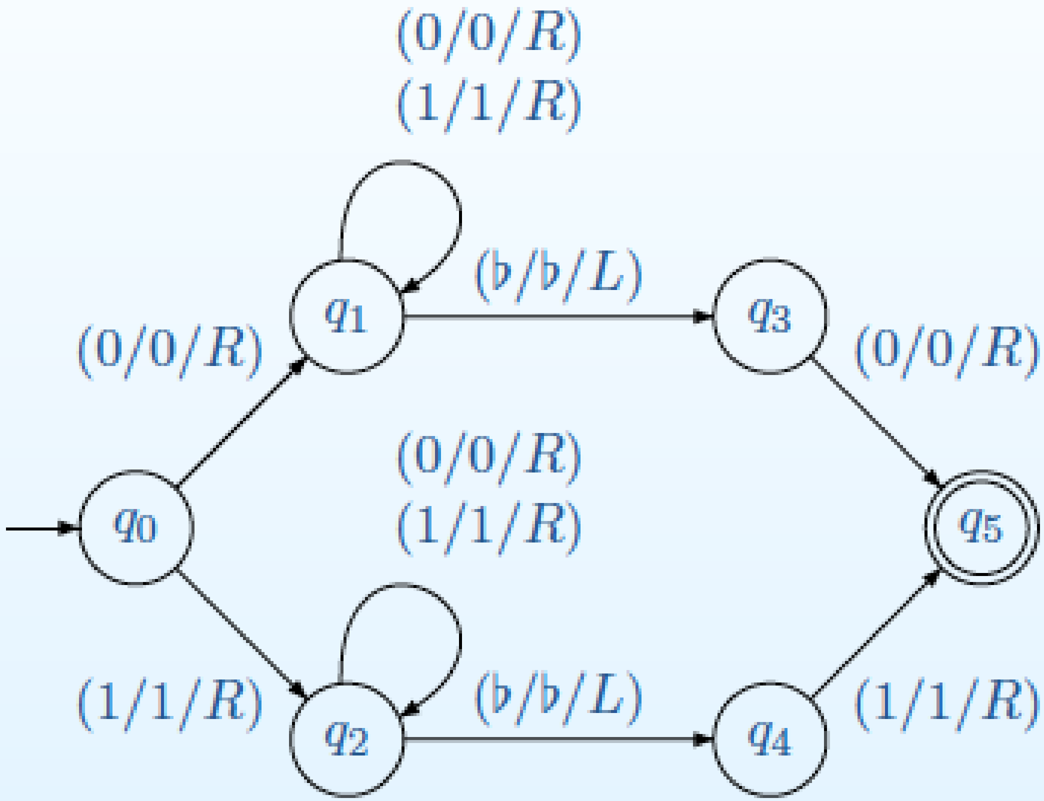
$F = \{q_2\}$

$$\left|\begin{array}{ccccc} 0 & 0 & 1 & 1 & \flat \\ \uparrow & & & & \\ q_0 & & & & \end{array}\right| \Rightarrow \left|\begin{array}{ccccc} 0 & 0 & 1 & 1 & \flat \\ & \uparrow & & & \\ & q_0 & & & \end{array}\right| \Rightarrow \left|\begin{array}{ccccc} 0 & 0 & 1 & 1 & \flat \\ & & \uparrow & & \\ & & q_0 & & \end{array}\right| \Rightarrow \left|\begin{array}{ccccc} 0 & 0 & 1 & 1 & \flat \\ & & & \uparrow & \\ & & & q_0 & \end{array}\right|$$

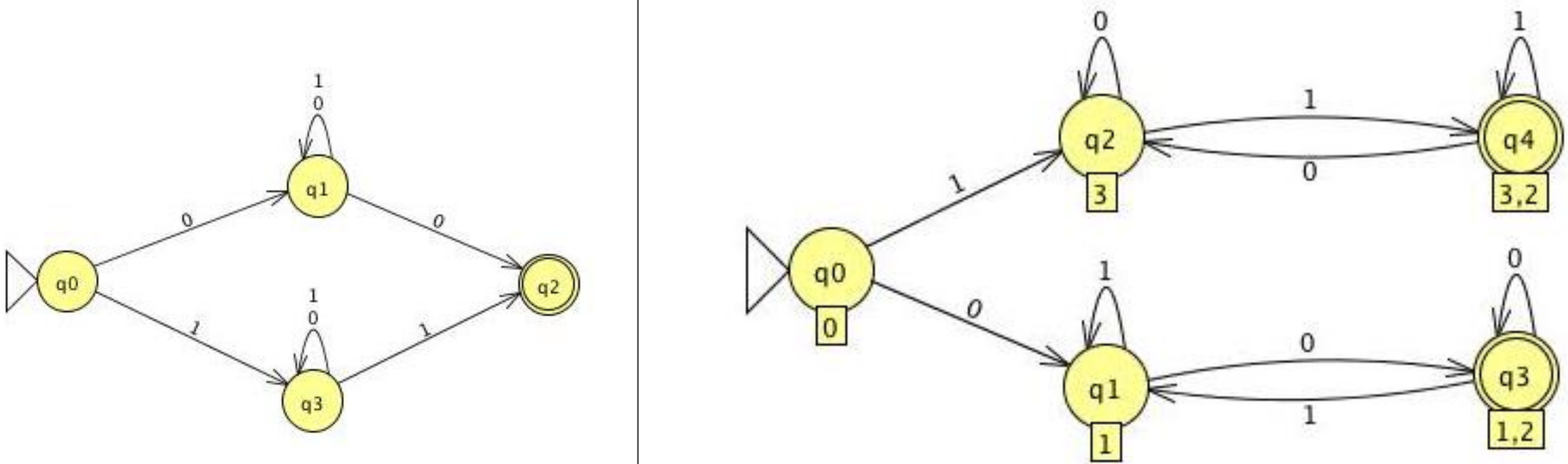
$$\left|\begin{array}{ccccc} 0 & 0 & 1 & 1 & \flat \\ & & & & \uparrow \\ & & & & q_0 \end{array}\right| \Rightarrow \left|\begin{array}{ccccc} 0 & 0 & 1 & 1 & \flat \\ & & & \uparrow & \\ & & & q_1 & \end{array}\right| \Rightarrow \left|\begin{array}{ccccc} 0 & 0 & 1 & 1 & \flat \\ & & & & \uparrow \\ & & & & q_2 \end{array}\right|$$

# Máquina de Turing como *aceptora*

Ejemplo 2: Máquina que acepta el lenguaje de palabras sobre $\{0,1\}$ que comienzan y acaban con el mismo símbolo



# Autómata Finito del ejemplo anterior



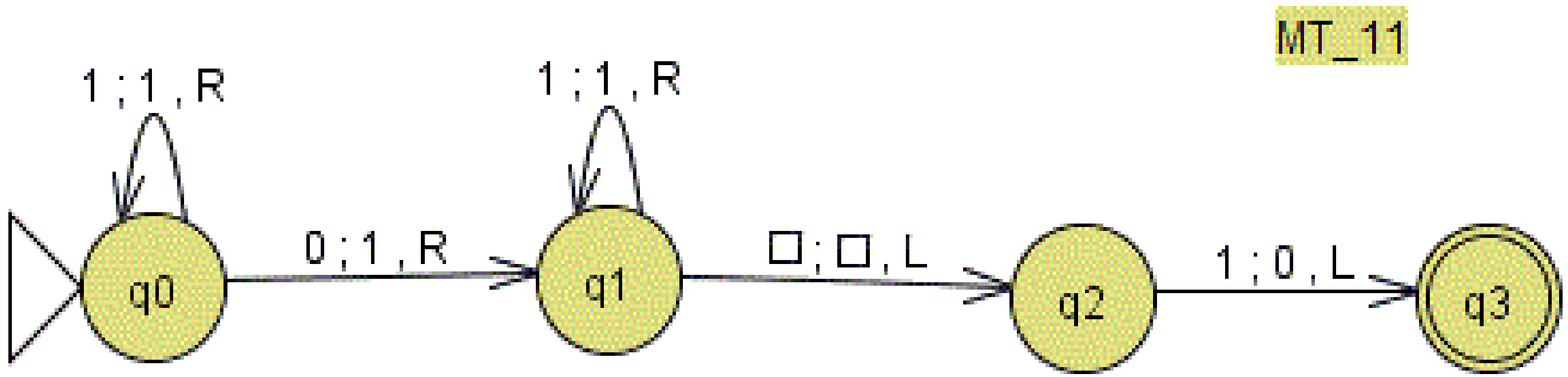

AFN (Izquierda) y equivalente AFD

# Problemas

1. MT que, dados dos enteros x e y, calcule x+y:
   - Ejemplo de contenido inicial de la cinta: w(x)0w(y)
   - Ejemplo de contenido final de la cinta: w(x+y)0

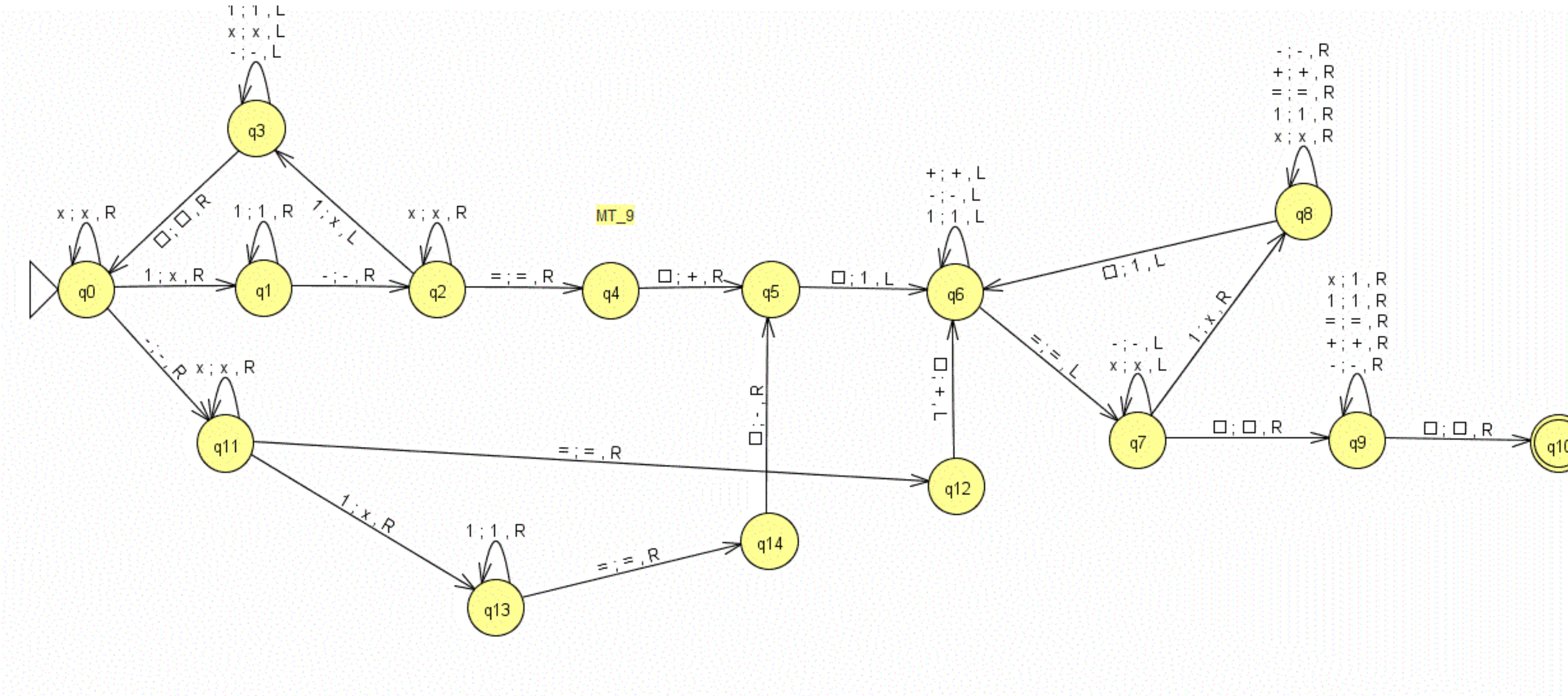
2. MT que realice la resta de dos números en alfabeto unario:
   - Ejemplo de contenido inicial de la cinta: “111-11=”.
   - Ejemplos de contenido final de la cinta: “1111-11=+11”, “11-1111=-11”, “11-11=+”.

# Problemas




MT que computa x+y

# Problemas

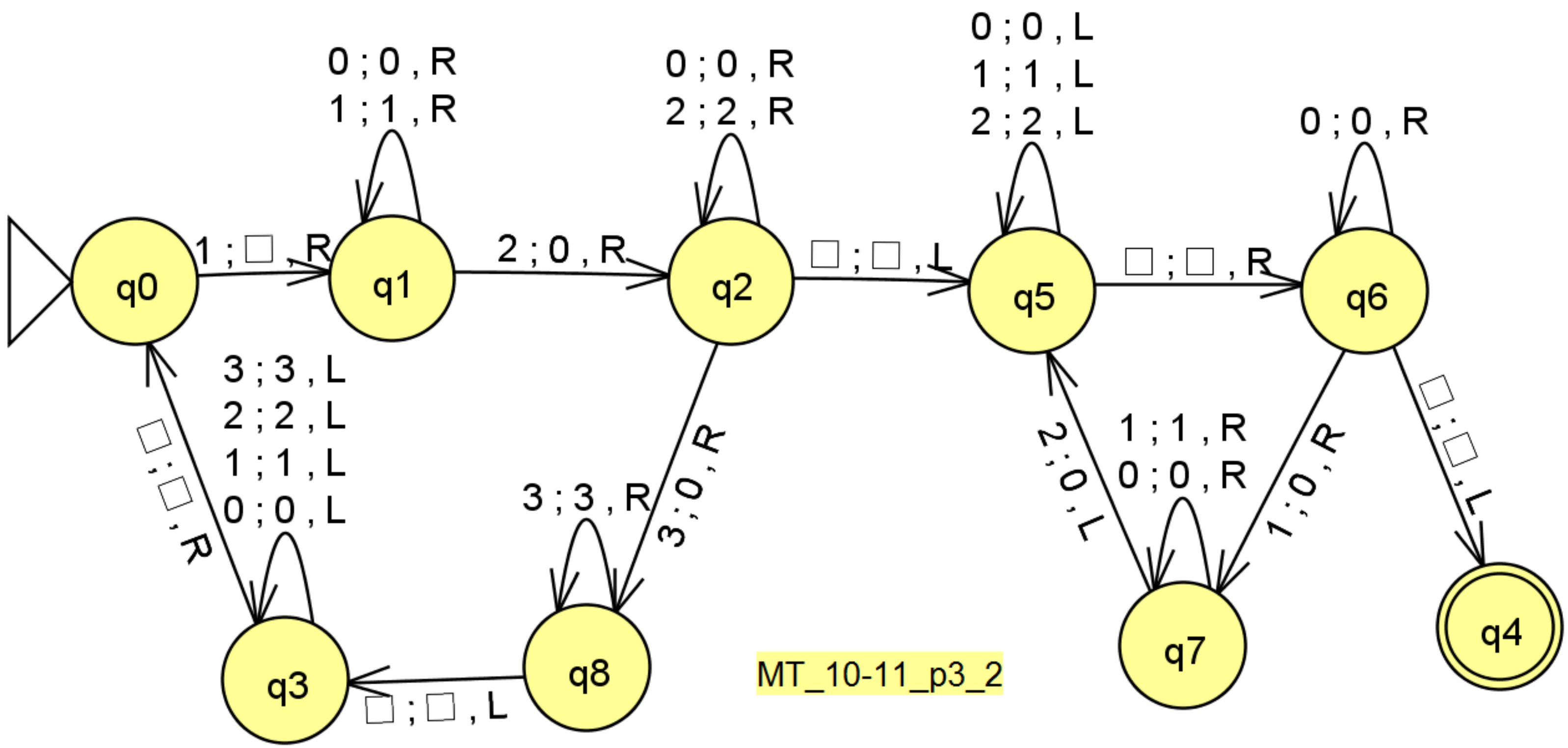



MT que realiza la resta

# Problemas

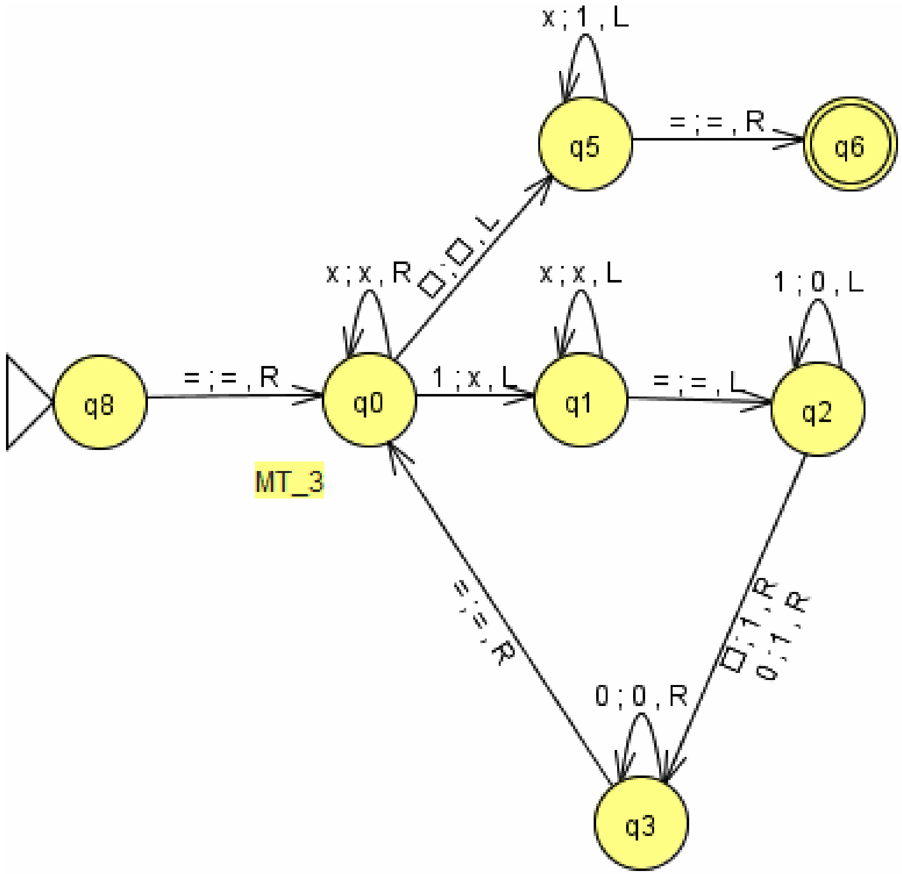
1. MT que reconozca expresiones del siguiente lenguaje: $L = \{1^n\ 2^n\ 3^k \mid n > k\}$
2. MT que, dados dos números enteros (a, b), acepte cuando $a \geq b$.
3. MT que convierta un número entero en formato unario a formato binario.

# Problemas




MT asociada al lenguaje: L = $\{1^n\ 2^n\ 3^k \mid n > k\}$

# Problemas




MT que transforma números unarios en binarios

# Máquina de Turing (MT)

- La MT finaliza el procesamiento cuando llega a un estado de parada
  - No hay transiciones definidas para esa combinación de estado y símbolo
  - Se asume que los estados finales no tienen transiciones definidas
    - Una MT se parará siempre que alcance un estado final

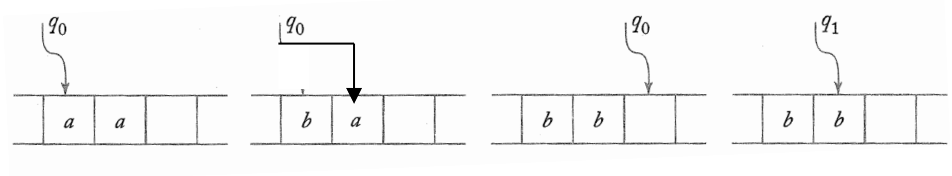
- En una MT **no es necesario leer todo el contenido** de la cinta para aceptar
  - Los AF y AP sí requieren leer toda la entrada

# Máquina de Turing

- Ejemplo:

$$Q = \{q_o, q_1\}, \quad \Sigma = \{a, b\}, \quad \Gamma = \{a, b, \Box\}, \quad F = \{q_1\},$$

$$\delta(q_0, a) = (q_0, b, R), \quad \delta(q_0, b) = (q_0, b, R), \quad \delta(q_0, \Box) = (q_1, \Box, L).$$

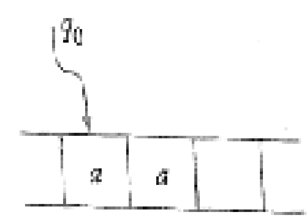



- Ejemplo: no hay parada de la MT

$$Q = \{q_o, q_1\}, \quad \Sigma = \{a, b\}, \quad \Gamma = \{a, b, \Box\}, \quad F = \{\ \},$$

$$\delta(q_0, a) = (q_1, a, R), \quad \delta(q_0, b) = (q_1, b, R), \quad \delta(q_0, \Box) = (q_1, \Box, R),$$

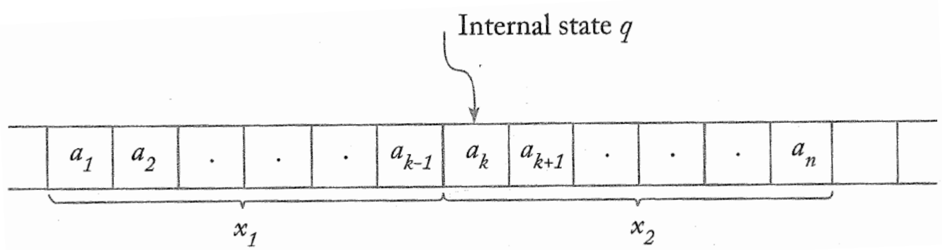
$$\delta(q_1, a) = (q_0, a, L), \quad \delta(q_1, b) = (q_0, b, L), \quad \delta(q_1, \Box) = (q_0, \Box, L).$$



# Máquina de Turing

- Resumen de las características de la MT estándar
  - Cinta infinita en ambas direcciones
  - La función de transición es determinista
    - Máximo un movimiento definido para cada configuración
  - No hay una entrada ni una salida específicas: codificados en la cinta

- Descripción instantánea
  - Estado
  - Contenido de la cinta
  - Posición de la cabeza de lectura/escritura
  - $a_1 \ldots a_{k-1} q a_k \ldots a_n$




  - Blancos sólo incluidos si son relevantes

# Máquina de Turing

Sea M $= (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

Descripción instantánea de M: $a_1 \ldots a_{k-1}q_1a_k \ldots a_n$, $a_i \in \Gamma$, $q_1 \in Q$

Movimiento a la derecha:
$a_1\ldots a_{k-1}q_1a_k\ldots a_n$ |- $a_1 \ldots a_{k-1}bq_2a_{k+1}\ldots a_n$, si $\delta(q_1, a_k) = (q_2, b, D)$

Movimiento a la izquierda:
$a_1\ldots a_{k-1}q_1a_k\ldots a_n$ |- $a_1\ldots a_{k-2}q_2a_{k-1}ba_{k+1}\ldots a_n$, si $\delta(q_1, a_k) = (q_2, b, I)$

M se para partiendo de $x_1q_ix_2$, si $x_1q_ix_2$ |-* $y_1q_jay_2$, para algún $\delta(q_j, a)$ no definido

- Computación: secuencia de configuraciones que llevan a la parada (entre ellas alcanzar un estado final, claro)
- No parada: $x_1q_ix_2$ |-* $\infty$

# Máquina de Turing y Lenguajes

- $L(M) = \{w \in \Sigma^+ : q_0 w \mid\!-^* x_1 q_f x_2, q_f \in F, x_1, x_2 \in \Gamma^*\}$

- Los espacios en blanco se usan para delimitar la cadena de entrada
  - La cadena vacía no forma parte del lenguaje para poder limitar la región en la que se busca la entrada.

- Si $w \notin L(M)$:
  - M se para en un estado no final
  - M entra en un bucle infinito: no hay parada

- Ejemplo: MT que reconoce $L = \{0^n 1^n \mid n \geq 1\}$

- Una TM que acepta: $\{0^n1^n | n \geq 1\}$
- La podemos definir como: $M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, 1, X, Y, B\}, \delta, q_0, B, \{q_4\})$
- Con la siguiente tabla de transición:

| | Símbolo | | | | |
|---|---|---|---|---|---|
| Estado | 0 | 1 | X | Y | B |
| $q_0$ | $(q_1, X, R)$ | $-$ | $-$ | $(q_3, Y, R)$ | $-$ |
| $q_1$ | $(q_1, 0, R)$ | $(q_2, Y, L)$ | $-$ | $(q_1, Y, R)$ | $-$ |
| $q_2$ | $(q_2, 0, L)$ | $-$ | $(q_0, X, R)$ | $(q_2, Y, L)$ | $-$ |
| $q_3$ | $-$ | $-$ | $-$ | $(q_3, Y, R)$ | $(q_4, B, R)$ |
| $q_4$ | $-$ | $-$ | $-$ | $-$ | $-$ |

# Máquina de Turing

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

# Máquina de Turing: L = $\{0^n 1^n \mid n \geq 1\}$

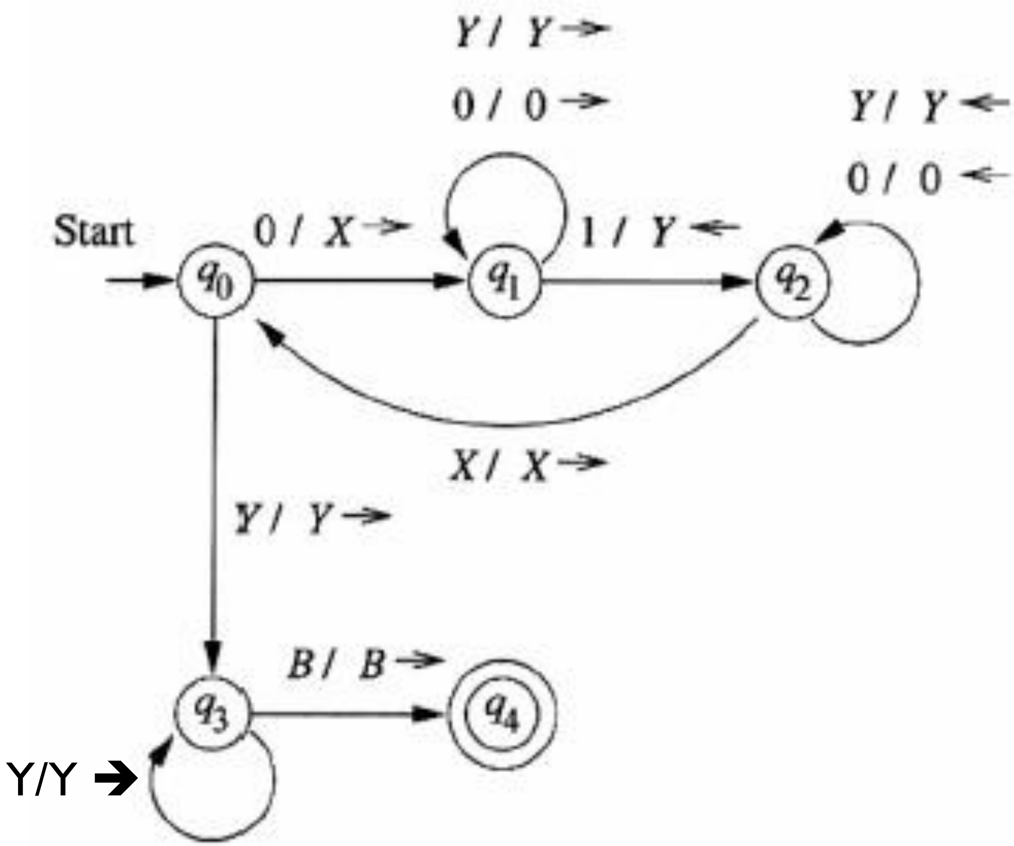


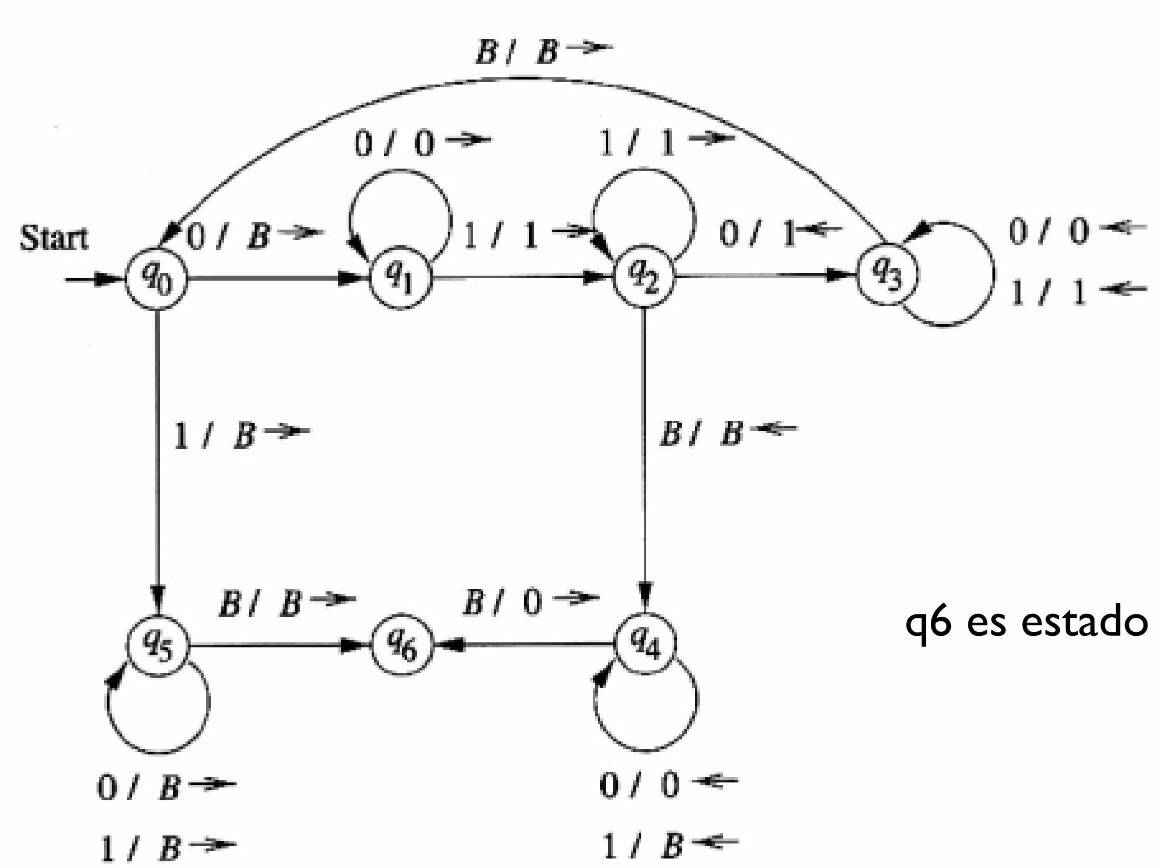

Diagrama de transición del ejemplo

- Diseñar una TM que calcula la función $\dot{-}$ llamada *monus* o substracción propia, que se define como: $m \dot{-} n = max(m-n, 0)$.
- La siguiente tabla y diagrama de transición lo definen, con entrada $0^m 10^n$:

| Estado | Símbolo 0 | 1 | B |
|---|---|---|---|
| $q_0$ | $(q_1, B, R)$ | $(q_5, B, R)$ | $-$ |
| $q_1$ | $(q_1, 0, R)$ | $(q_2, 1, R)$ | $-$ |
| $q_2$ | $(q_3, 1, L)$ | $(q_2, 1, R)$ | $(q_4, B, L)$ |
| $q_3$ | $(q_3, 0, L)$ | $(q_3, 1, L)$ | $(q_0, B, R)$ |
| $q_4$ | $(q_4, 0, L)$ | $(q_4, B, L)$ | $(q_6, 0, R)$ |
| $q_5$ | $(q_5, B, R)$ | $(q_5, B, R)$ | $(q_6, B, R)$ |
| $q_6$ | $-$ | $-$ | $-$ |

# Máquina de Turing | Función *monus*

# Máquina de Turing: ejemplo




q6 es estado final

Diagrama de transición de la función *monus*

# Computación de funciones

- Una función ***f*** con dominio ***D*** es Turing-computable o computable sin más, si existe una MT M = (Q, Σ, Γ, δ, $q_0$, B, F) tal que:

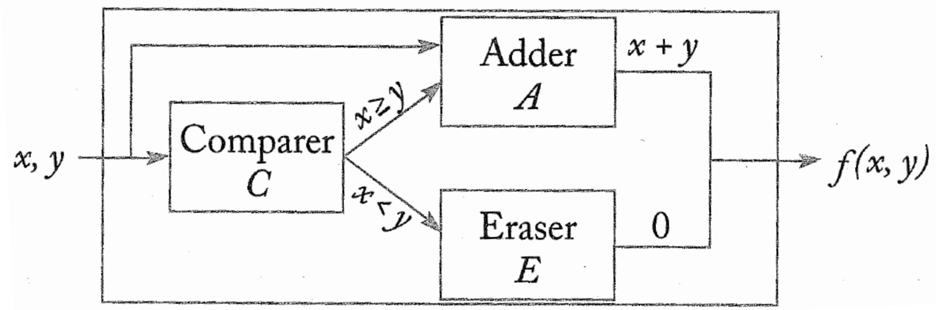
  $q_0w$ |-* $q_f$***f***(w), $q_f \in$ F, para todo w $\in$ ***D***

- Todas las funciones matemáticas comunes, no importa lo complicadas que sean, son Turing-computables

- Ejemplos:
  - MT que, dados dos enteros x e y en alfabeto unario, compute x+y
    - Contenido de la cinta: w(x)0w(y) |-* w(x+y)0
  - MT que duplique cadenas de 1's
  - MT que realice la resta de dos números en alfabeto unario
    - Ejemplo de contenido inicial de la cinta: "111-11="
    - Ejemplos de contenido final de la cinta: "1111-11=+11", "11-1111=-11", "11-11=+"

# Combinación de MT

- Ejemplo: f(x, y) = x + y   si $x \geq y$; f(x, y) = 0  si $x < y$




  - Comparador:
    - $q_{C,0}w(x)0w(y) \mid\!-^* q_{A,0}w(x)0w(y)$  si $x \geq y$
    - $q_{C,0}w(x)0w(y) \mid\!-^* q_{E,0}w(x)0w(y)$  si $x < y$
  - Sumador:
    - $q_{A,0}w(x)0w(y) \mid\!-^* q_{A,f}w(x+y)$
  - Borrador:
    - $q_{E,0}w(x)0w(y) \mid\!-^* q_{E,f}0$

# Tesis de Church-Turing

- Hipótesis: cualquier problema de decisión resoluble puede ser transformado en un problema equivalente para una MT

- Argumentos:
  - Cualquier problema que se pueda resolver en una computadora también se puede resolver con una MT
  - No se ha encontrado ningún problema resoluble (por un algoritmo) para el que no se pueda escribir un programa para una MT
  - Se han propuesto modelos alternativos de computación, pero ninguno ha probado ser más potente que el modelo de MT

# Tesis de Church-Turing

- **Definición de algoritmo**: un algoritmo para una función f: D $\rightarrow$ R es una MT, la cual dada cualquier entrada d $\in$ D en su cinta, finalmente se para con la respuesta correcta f(d) $\in$ R en la cinta:

$$q_0 d \mid\!-^* q_f f(d),\ q_f \in F \ \text{ y para todo } \ d \in D$$

  - Usando la tesis de Church-Turing, podemos sustituir en la definición "MT" por "programa en Java", "programa en C", etc.

# Otros modelos de MT

- Pequeñas variaciones:
  - MT con opción de no-movimiento
  - MT con cinta semiinfinita
  - MT con cinta de entrada
- Almacenamiento más complejo
  - MT multicinta
  - MT multidimensionales
- MT no deterministas
- Y más…

# MT con opción de no-movimiento

- $\delta$: Q x $\Gamma \rightarrow$ Q x $\Gamma$ x {I, D, E}
  - E indica que la cabeza de lectura/escritura permanece estática

- Teorema: la clase de MT con opción de no-movimiento es equivalente a la clase de MT estándar

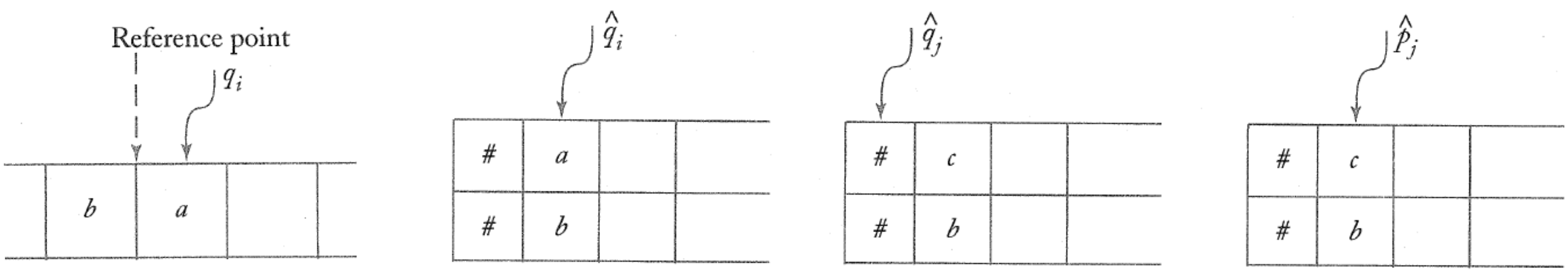
- Simulación de una MT con opción de no-movimiento ($\delta$) con una MT ($\delta'$) :
  - Por cada $\delta(q_i, a) = (q_j, b, I \text{ o } D)$, se incluye $\delta'(q_i, a) = (q_j, b, I \text{ o } D)$
  - Por cada $\delta(q_i, a) = (q_j, b, E)$, se incluyen $\delta'(q_i, a) = (q_{jS}, b, D)$ y $\delta'(q_{jS}, c) = (q_j, c, I)$ (una transición por cada $c \in \Gamma$)

# MT con cinta semiinfinita

- MT cuya cinta está limitada por un extremo
- Simulación de una MT estándar M por medio de una MT con cinta semiinfinita P:
  - Cinta de P con dos pistas
    - Pista superior: contenido de la cinta de M a la derecha de la referencia (situación inicial de la cabeza de M)
    - Pista inferior: contenido de la cinta de M a la izquierda de la referencia y en orden inverso
  - Los estados de P se dividen en dos conjuntos: un conjunto trabaja con la pista superior y otro con la inferior
  - Unos marcadores especiales (#) en el extremo izquierdo permiten cambiar de pista

# MT con cinta semiinfinita

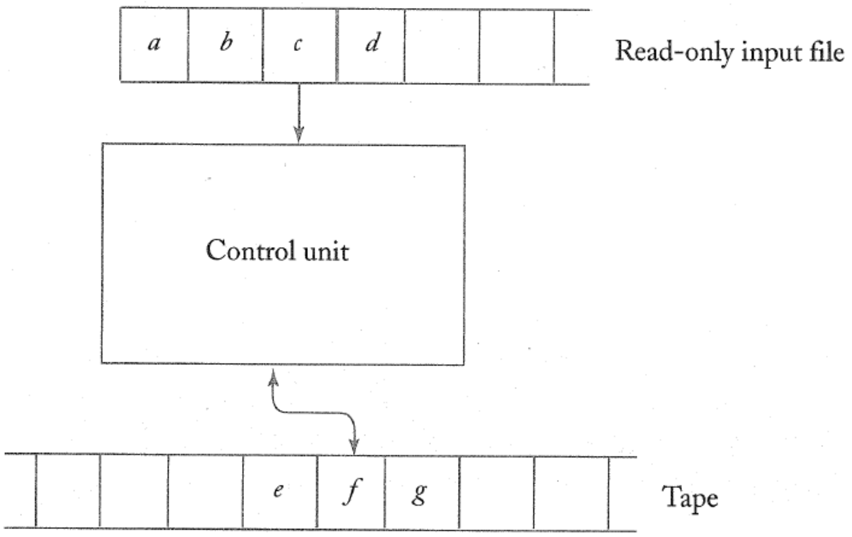
- Ejemplo:
  - $\delta(q_i, a) = (q_j, c, I)$ es simulado con
  - $\delta'(q'_i, (a, b)) = (q'_j, (c, b), I)$ y $\delta'(q'_j, (\#, \#)) = (p'_j, (\#, \#), D)$
    - $\{q'_i, q'_j\} \in Q_U$ y $p'_j \in Q_L$



# MT con cinta de entrada

- La entrada está escrita en una cinta de sólo lectura

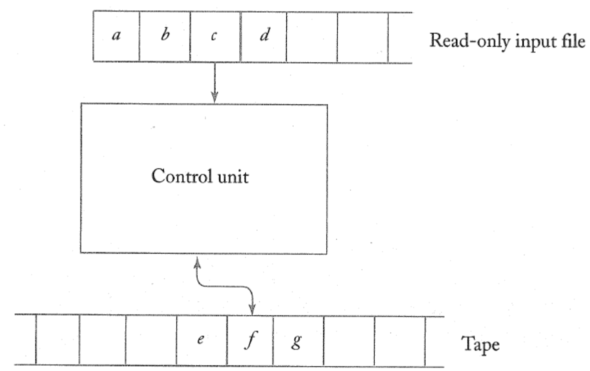
- Las transiciones se realizan en función del estado, el símbolo leído de la entrada y el símbolo leído por la cabeza de lectura/escritura en la cinta

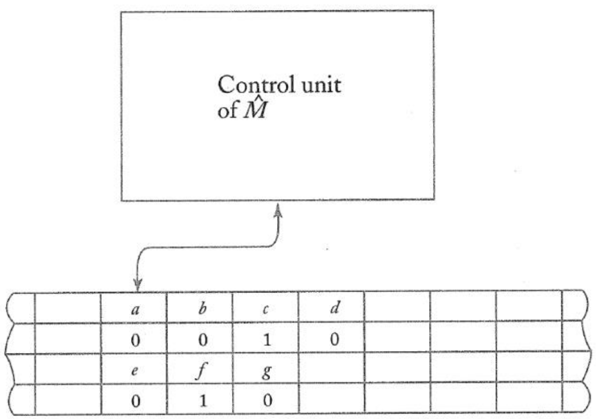



- Simulación de una MT con una MT con cinta de entrada
  - Copiar el contenido de la cinta de entrada en la cinta

# MT con cinta de entrada

- Simulación de una MT con cinta de entrada $M_1$ con una MT $M_2$
  - Cinta con 4 pistas: valores de entrada, posición de la cabeza de lectura, contenido de la cinta y posición de la cabeza de lectura/escritura
  - Ejemplo:



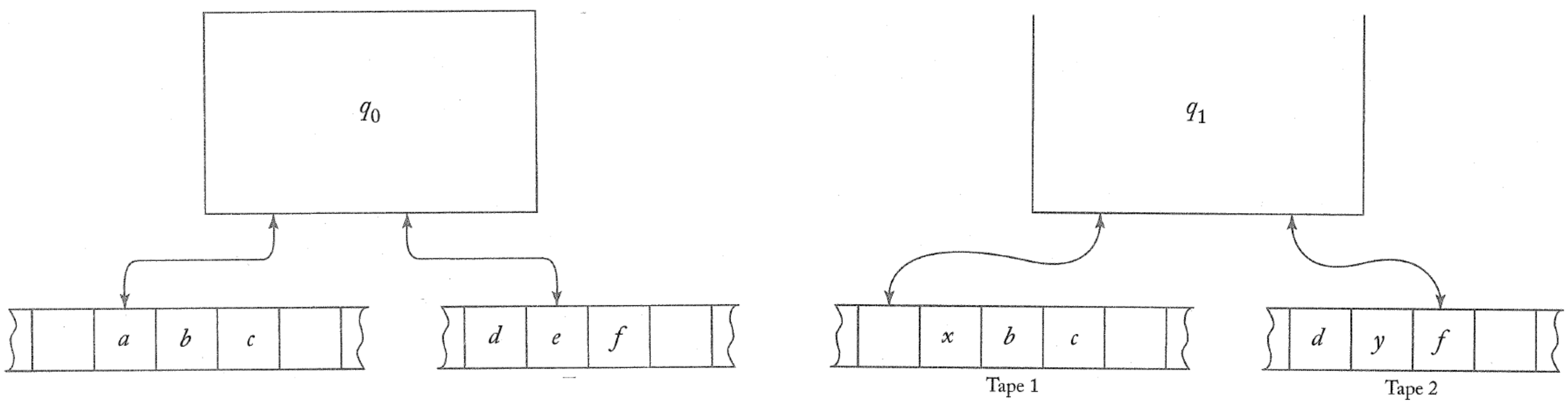




  - La simulación requiere varios movimientos en $M_2$ por cada movimiento en $M_1$
    - Posición de partida: extremo izquierdo de la cinta
    - Búsqueda de la posición de la cabeza de lectura en la pista 2
    - Lectura del símbolo correspondiente en la pista 1 y transición de estado
    - Búsqueda de la posición de la cabeza de lectura/escritura en la pista 4
    - Lectura del símbolo correspondiente en la pista 3 y transición de estado
    - Modificación de las pistas para representar el movimiento en $M_1$
    - Vuelta a la posición de partida para simular el siguiente movimiento

# MT multicinta

- $\delta: Q \times \Gamma^n \to Q \times \Gamma^n \times \{I, D\}^n$

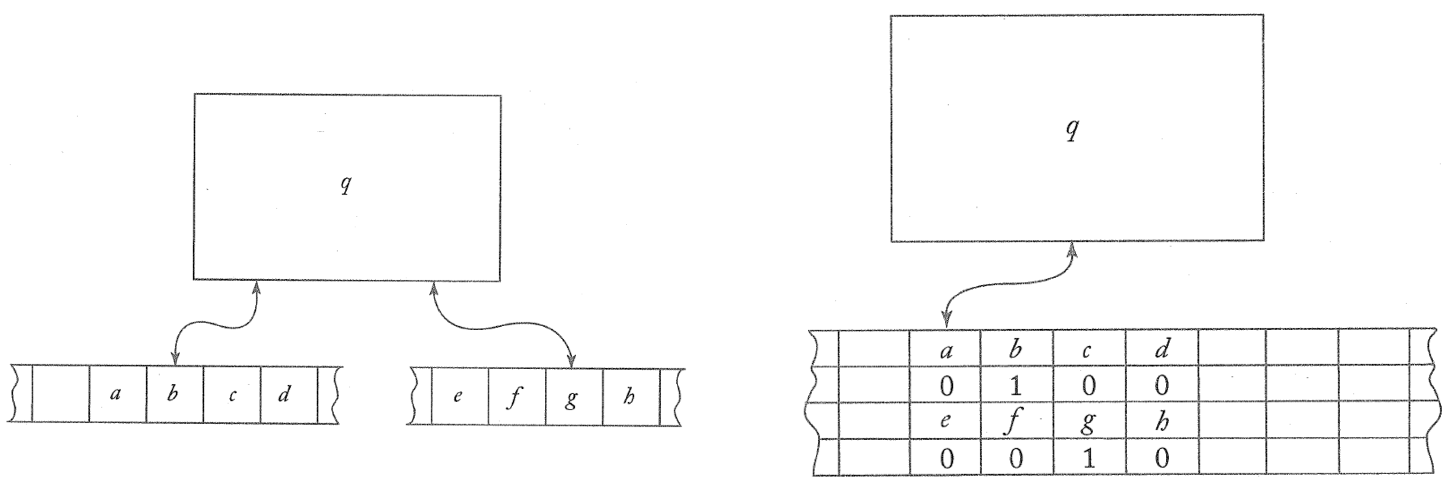
- Ejemplo: $\delta(q_0, a, e) = (q_1, x, y, I, D)$



# MT multicinta

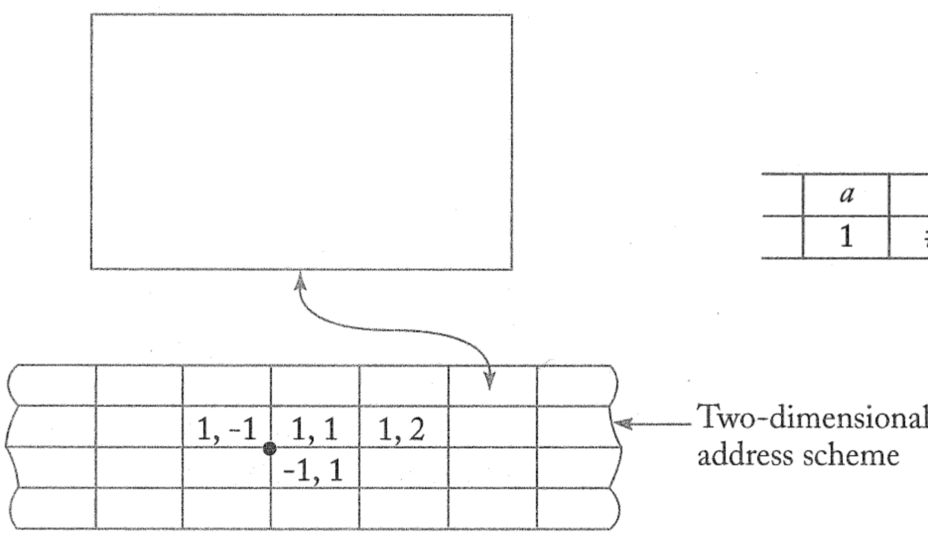
- Simulación de una MT multicinta con una MT: uso de 2n pistas (n es el número de cintas de la MT multicinta)
    - Pistas impares: representan el contenido de las cintas
    - Pistas pares: representan la posición de la cabeza en las cintas
    - Los pasos a ejecutar son similares al de la simulación de MT con cinta de entrada

- Ejemplo:



# MT multidimensional

- La cinta es infinita en más de una dimensión
- MT bidimensional: $\delta$: Q x $\Gamma \rightarrow$ Q x $\Gamma$ x {I, D, AR, AB}
- Simulación de una MT bidimensional con una MT: cinta con dos pistas
  - Pista 1: almacena el contenido de la cinta bidimensional
  - Pista 2: contiene las direcciones asociadas al contenido de la pista 1
  - Para simular un movimiento, se busca en la pista 2 la dirección de la celda a la que se debe desplazar la cabeza de lectura/escritura




| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| *a* | | | | *b* | | | | | |
| 1 | # | 2 | # | 1 | 0 | # | - | 3 | # |

# MT no determinista

- $\delta$: $Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{I, D\}}$
  - Ejemplo: $\delta(q_0, a) = \{(q_1, b, D), (q_2, c, I)\}$
- Una MT no determinista puede verse como una MT que puede replicarse a sí misma cuando sea necesario
  - Por cada posible transición se crea una réplica
- Simulación de una MT no determinista con una MT
  - Se crea una cinta con 2n pistas, donde n es el número de máquinas a simular
  - Cada nueva MT implica la inicialización de dos nuevas pistas
  - Una pista representa el contenido de la cinta y la otra el estado de la MT

# MT no determinista

- <u>Ejemplo</u>: $\delta(q_0, a) = \{(q_1, b, D), (q_2, c, I)\}$

| | | | | | |
|---|---|---|---|---|---|
| | # | # | # | # | # |
| | # | $a$ | $a$ | $a$ | # |
| | # | $q_0$ | | | # |
| | # | # | # | # | # |
| | | | | | |
| | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| # | # | # | # | # | # |
| # | | $b$ | $a$ | $a$ | # |
| # | | | $q_1$ | | # |
| # | | $c$ | $a$ | $a$ | # |
| # | $q_2$ | | | | # |
| # | # | # | # | # | # |

- MT deterministas y no deterministas son equivalentes

# MT universal (MTU)

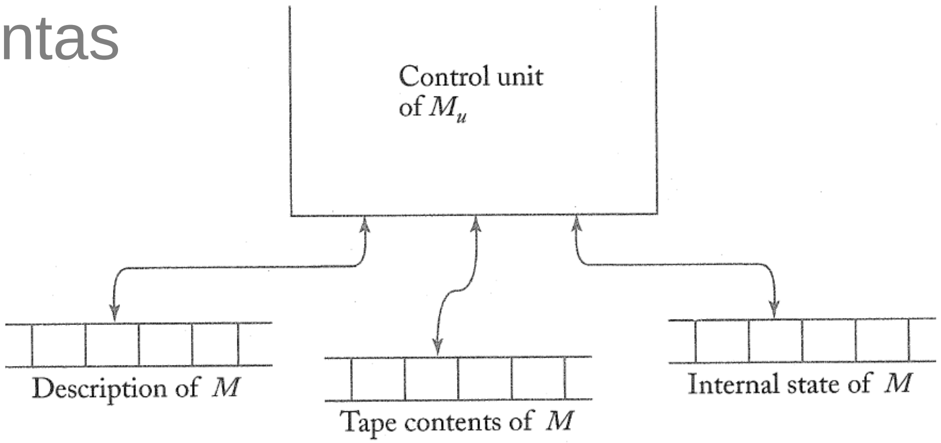
- Las MTU son reprogramables
- Dada una descripción de cualquier MT M y una cadena w, una MTU puede simular la computación de M para w

- Descripción de una MT:
  - $Q = \{q_1, q_2, \ldots, q_n\}$, siendo $q_1$ el estado inicial y $q_n$ el final
  - $\Gamma = \{a_1, a_2, \ldots, a_m\}$, siendo $a_1$ un espacio en blanco

- Codificación de una MT
  - $q_1 = 1$ $q_2 = 11, \ldots$
  - $a_1 = 1$ $a_2 = 11, \ldots$
  - I = 1, D = 11
  - 0 es el símbolo separador
  - Ejemplo: $\delta(q_1, a_2) = (q_2, a_3, I)$ se codifica como 10110110111010

# MT universal (MTU)

- Una MTU tiene tres cintas




- Funcionamiento:
  - Se examina el contenido de las cintas 2 y 3: configuración de M
  - Se consulta la cinta 1 para determinar la transición a realizar
  - Se modifican las cintas 2 y 3 como resultado del movimiento realizado

- Una MT, dado cualquier programa, puede realizar las computaciones especificadas y es, por tanto, un modelo adecuado de una computadora de propósito general
- Cualquier MT puede ser codificada con ceros y unos

# Gramáticas sensibles al contexto

Gramáticas sin restricciones (GSR), G = (NT, T, S, P)

- Producciones:

$$x \rightarrow y$$
$$x \in (NT/T)^{+}$$
$$y \in (NT/T)^{*}$$

- Las GSR generan los LRE

Gramáticas sensibles al contexto (GSC), G = (NT, T, S, P)

- Producciones:

$$\alpha A \beta \rightarrow \alpha \gamma \beta$$

donde:

- A es un símbolo no terminal.
- α,β son cadenas de símbolos terminales y no terminales (pueden ser la cadena vacía)
- γ es una cadena no vacía de símbolos terminales y/o no terminales, igual o más larga que A

- Un lenguaje L es sensible al contexto (LSC) si existe una GSC tal que L = L(G) o L = L(G) ∪ {λ}

# Lenguajes sensibles al contexto

- Ejemplo: GSC que genera L = $\{a^n b^n c^n : n \geq 1\}$

$$
\begin{aligned}
S &\to abc \mid aAbc, \\
Ab &\to bA, \\
Ac &\to Bbcc, \\
bB &\to Bb, \\
aB &\to aa \mid aaA.
\end{aligned}
$$

$$
\begin{aligned}
S &\Rightarrow aAbc \Rightarrow abAc \Rightarrow abBbcc \\
&\Rightarrow aBbbcc \Rightarrow aaAbbcc \Rightarrow aabAbcc \\
&\Rightarrow aabbAcc \Rightarrow aabbBbccc \\
&\Rightarrow aabBbbccc \Rightarrow aaBbbbccc \\
&\Rightarrow aaabbbccc.
\end{aligned}
$$

- Los LSC (que no contengan $\lambda$) son reconocidos por los Autómatas Linealmente Acotados (ALA)

- LSC $\subset$ LREC

# Autómatas linealmente acotados (ALA)

- Restricciones en el uso de la cinta
  - Uso de una parte finita de la cinta: autómata de estados finitos
  - Uso de la parte de la cinta ocupada por la cadena de entrada: ALA
    - Cuanta mayor longitud tenga la cadena de entrada, mayor es el espacio a utilizar

- Definición: un ALA es una **MT determinista** $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ con la restricción de que $\Sigma$ debe contener dos símbolos especiales:
  - [: marcador izquierdo, con transiciones del tipo $\delta(q_i, [) = (q_j, [, D)$
  - ]: marcador derecho, con transiciones del tipo $\delta(q_i, ]) = (q_j, ], I)$

- $L(M) = \{w \in \Sigma^+ : q_0[w] \mid\!-^* [x_1 q_f x_2], q_f \in F, x_1, x_2 \in \Gamma^*\}$

- Los ALA son más potentes que los AP y menos que las MT

# Lenguajes recursivos y recursivamente enumerables

Un lenguaje L es **recursivamente enumerable** (LRE) si existe una MT que acepta cualquier cadena del lenguaje y se para:

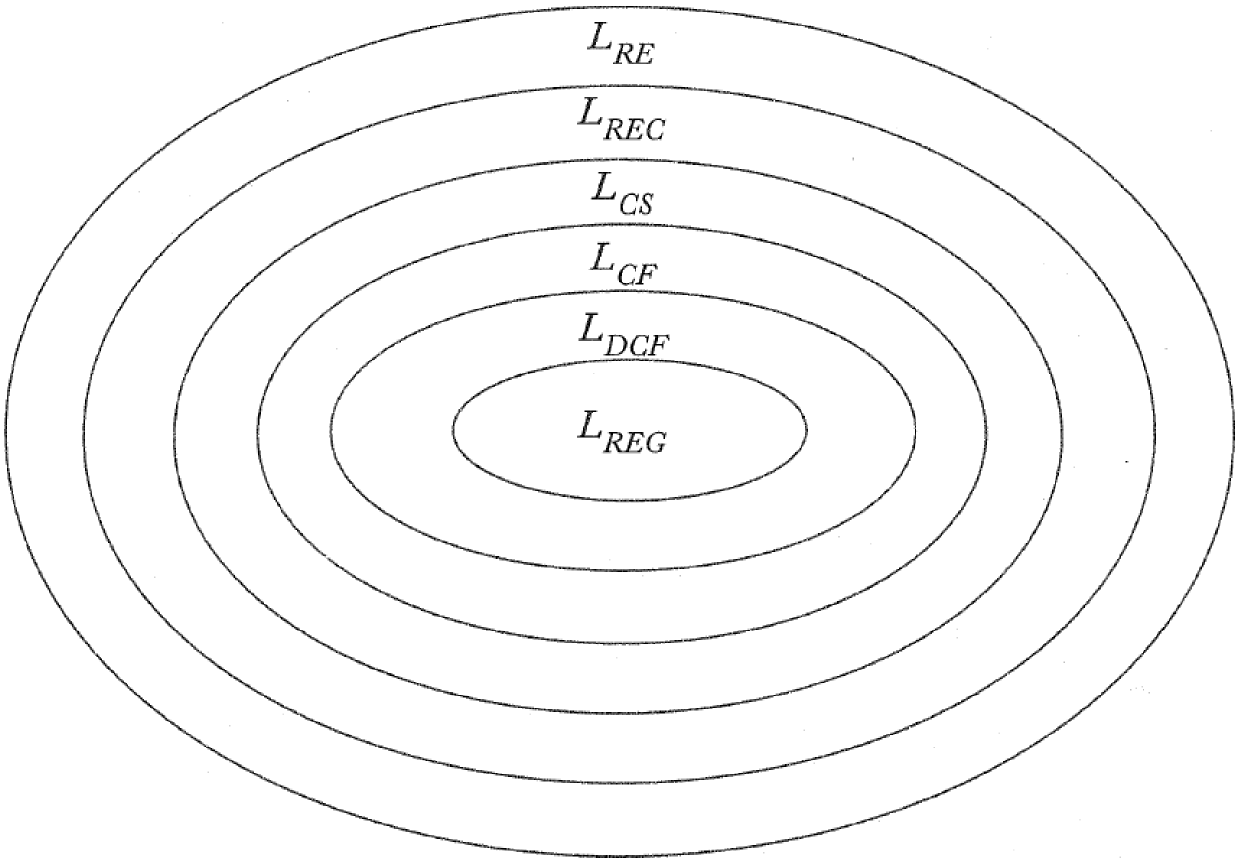
- Si $w \in L$, $q_0 w \vdash^* x_1 q_f x_2$ , $q_f \in F$

Sin embargo:

- Si $w \notin L$, la MT **se parará** en un estado no final –rechazándola- o **entrará en un bucle infinito**

Un lenguaje L sobre un alfabeto $\Sigma$ es **recursivo** (LRC) si existe una MT que acepta L y se para con cualquier cadena $w \in \Sigma^+$ **-se para la acepte o no**-

Hay lenguajes que no son LRE. La demostración es compleja, puesto que todos los lenguajes descritos de forma algorítmica son aceptados por MT (son por tanto LRE)




**$L_{REG}$**: Lenguaje Regular
**$L_{DCF}$**: Lenguaje Determinista Independiente de Contexto
**$L_{CF}$**: Lenguaje Independiente de Contexto
**$L_{CS}$**: Lenguaje Dependiente de Contexto
**$L_{REC}$**: Lenguaje Recursivo
**$L_{RE}$**: Lenguaje Recursivamente enumerable

# Lenguajes formales

## Jerarquía de Chomsky