Teoría de la Complejidad Computacional

Tema 2: Modelos de Computación
Máquinas de Turing

David Orellana Martín

Grupo de Investigación en Computación Natural
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

dorellana@us.es

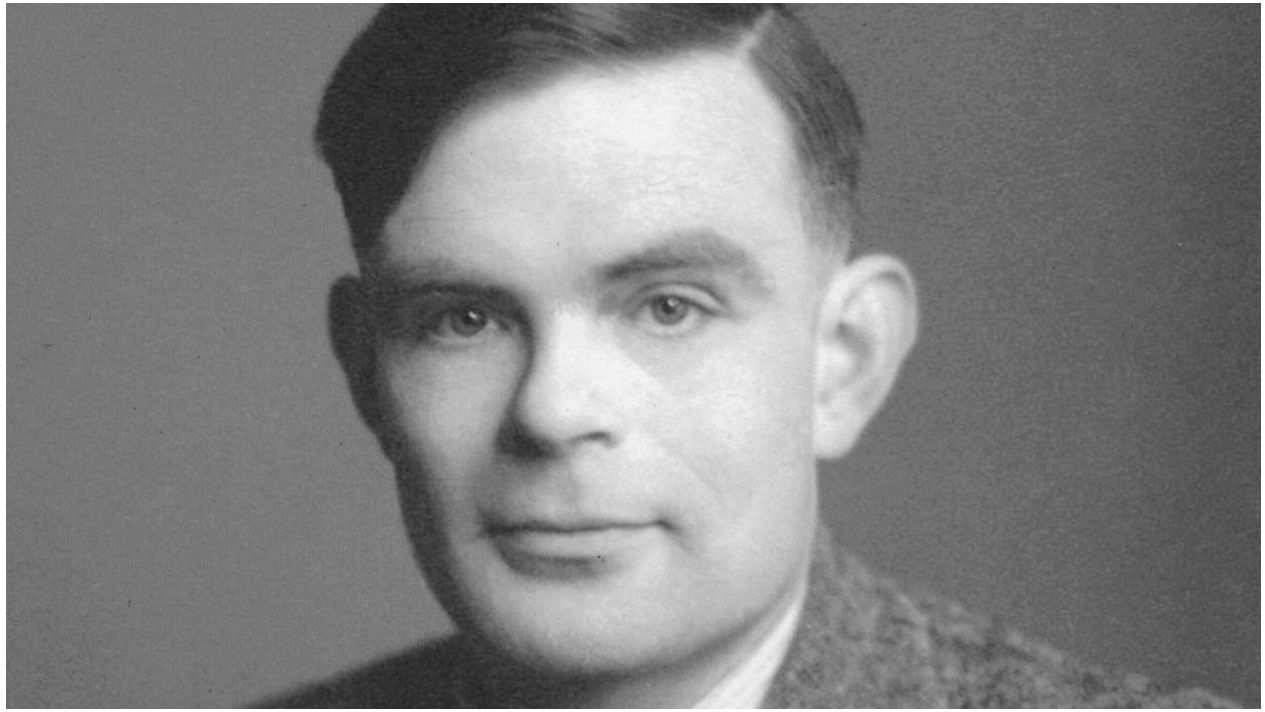
Máster Universitario en Matemáticas

Curso 2023-2024

Índice

* Alan Turing.
* Máquinas de Turing **deterministas**.
  - Sintaxis y Semántica.
  - MTDs de decisión.
  - MTDs que resuelven problemas de decisión.
  - MTDs que calculan funciones.
* Máquinas de Turing deterministas **con k cintas**.
* Máquinas de Turing **no deterministas**.

Alan Turing (1912-1954)



- ⋆ Nació el 23 de junio de 1912 en una zona residencial de Londres (Maida Vale).
- ⋆ Pionero en la introducción de modelos de computación (mayo de 1936).
- ⋆ Tesis doctoral: **Systems of logic based on ordinals** (mayo de 1938, dirigida por A. Church).

- ⋆ Al día siguiente de la entrada de Gran Bretaña en la Segunda Guerra Mundial, fue "reclutado" en Bletchley Park (septiembre de 1939), donde se encontraba el Servicio Británico de Descifrado.
- ⋆ Papel decisivo en esa guerra, descifrando información codificada por los alemanes (para transmitir órdenes a sus naves que operaban en el Atlántico).
- ⋆ Los alemanes codificaban la información usando la máquina **Enigma**.

- ⋆ A finales de 1939, A. Turing diseñó una máquina **electromecánica** de propósito específico: la máquina **Bombe** (mejorando una máquina desarrollada en 1938 por investigadores polacos).

- ⋆ Se estima que la aportación de A. Turing propició que la guerra finalizara unos **dos años antes**, evitando la muerte de varios millones de personas (algunos expertos lo estiman en unos 14 millones).

En 1950, A. Turing formuló una pregunta muy interesante: **Can machines think?**[1]

Para responder a esta cuestión formuló ésta otra pregunta **Are there imaginable digital computers which would do well in the imitation game**?

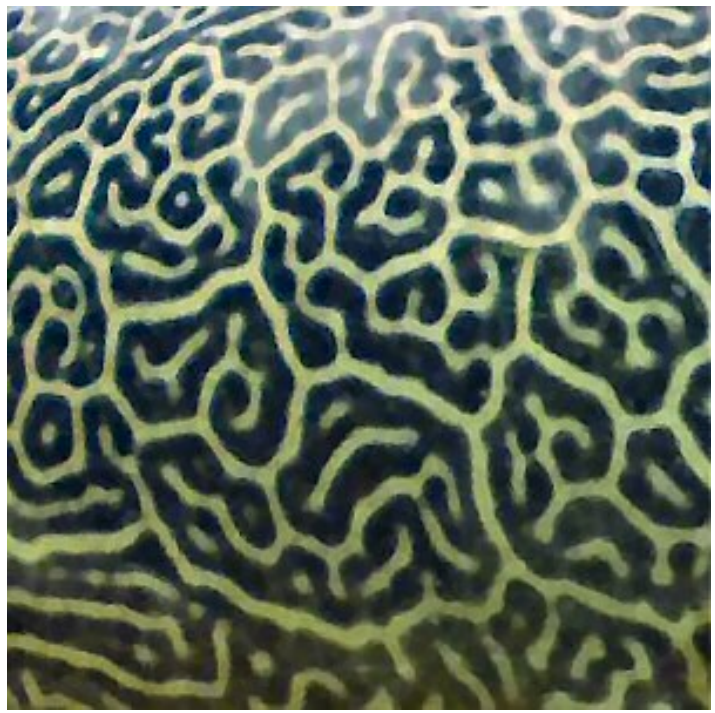
- ⋆ Conocido como **test de Turing**.
- ⋆ **Eugene Goostman** (desarrollado en 2001) una aplicación software "singular":

  - ∗ En 1950, A. Turing afirmó que en el año 2000 existirían máquinas capaces de mantener una conversación con una persona durante 5 m. usando mensajes de texto, sin que ésta supiera, con una probabilidad del 30%, si su interlocutor era o no una máquina.
  - ∗ Confundió a 45 de 155 personas encargadas de dictaminar si el "interlocutor" era una máquina.
  - ∗ Se considera la primera "máquina" que superó el test de Turing (en un polémico evento celebrado en 2014 por K. Warwick).

[1] A.M. Turing. Computing Machinery and Intelligence, **Mind**, Vol. **LIX**, Issue 236, October 1950, pp 433-460.

A. Turing es el pionero de una de las disciplinas científicas más relevante en la actualidad: la **Inteligencia Artificial**.

A partir de 1952, A.Turing se centró en el estudio de la **teoría de patrones**, aplicada al mecanismo de desarrollo de la forma[2] (**morfogénesis**).



A. Turing conjeturó la existencia de **patrones regulares** en el sistema biológico animal, a la hora de implementar la morfogénesis.

[2] A. Turing. The Chemical Basis of Morphogenesis. **Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences**, Vol. **237**, No 641. (Aug. 14, 1952), pp. 37-72.

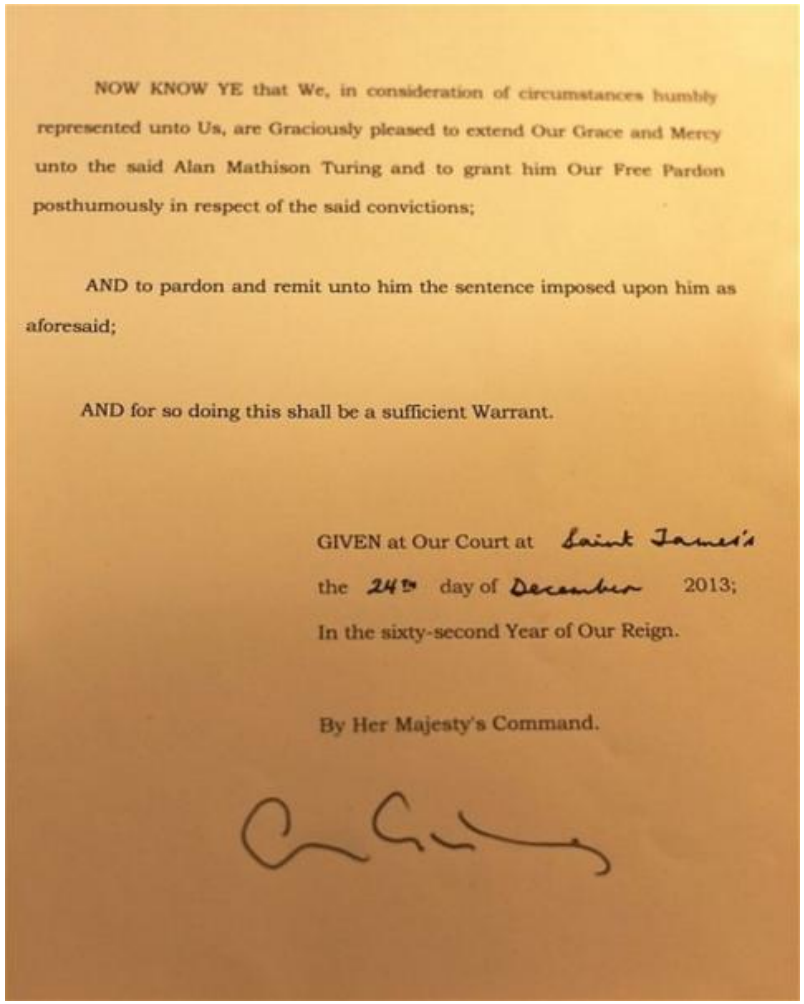
Morfogénesis

- Las células se organizan entre ellas para crear las diferentes estructuras y formas en los animales.
- Las estructuras resultantes son consecuencia de la **difusión** de moléculas por el espacio y las **reacciones** químicas que se produce entre ellas.
- La combinación de los **procesos de difusión/reacción** produce la creación de **patrones periódicos**.
  - La pigmentación de los animales (i.e. las rayas que aparecen regularmente en ciertos animales) eran causadas por dos sustancias químicas (**morfógenos**) que trabajaban, a la vez, como activadores e inhibidores.
  - Esta conjetura fue corroborada experimentalmente, en 2012, por unos investigadores del King's College, en Cambridge.

A. Turing era homosexual, fue juzgado por ello, y condenado por **gross indecency** (1952). Aceptó la castración química para evitar la entrada en prisión.

El **final**: 7 de junio de 1954 (presunto suicidio, al haber mordido una manzana rociada de cianuro).

Su rehabilitación pública (algo tardía) se produjo el día de Nochebuena de 2013.



NOW KNOW YE that We, in consideration of circumstances humbly represented unto Us, are Graciously pleased to extend Our Grace and Mercy unto the said Alan Mathison Turing and to grant him Our Free Pardon posthumously in respect of the said convictions;

AND to pardon and remit unto him the sentence imposed upon him as aforesaid;

AND for so doing this shall be a sufficient Warrant.

GIVEN at Our Court at Saint James's
the 24th day of December 2013;
In the sixty-second Year of Our Reign.

By Her Majesty's Command.

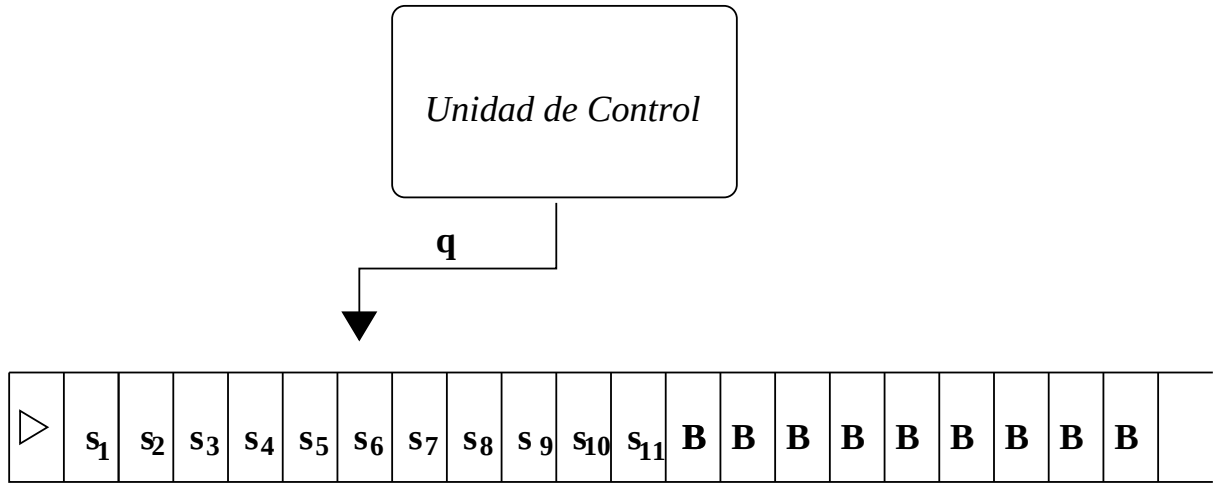
Máquina de Turing determinista: Sintaxis

Es una tupla, $M = (Q, \Sigma, q_0, F, \mathbf{B}, \triangleright, \delta)$:

- Q es un alfabeto finito (**estados**).
- Σ es un alfabeto finito (**de trabajo de la máquina**) tal que $\Sigma \cap Q = \emptyset$.
- $q_0 \in Q$ (**estado inicial**).
- $F = \{q_h, q_y, q_n\} \subseteq Q - \{q_0\}$ (**estados finales**, todos distintos entre sí).
- $\mathbf{B} \in \Sigma$ (**símbolo blanco**).
- $\triangleright \in \Sigma$ (**primer símbolo**) verificando que $\triangleright \neq \mathbf{B}$.
- δ es una aplicación (**función de transición**) de $(Q - F) \times \Sigma$ en $Q \times \Sigma \times \{\mathbf{L}, \mathbf{R}, \mathbf{S}\}$ tal que:
  - Si $\delta(q, \triangleright) = (q', s', x)$ entonces $s' = \triangleright \wedge x = \mathbf{R}$.

Informalmente, una máquina de Turing determinista M consta de:

* Una **cinta infinita**: celdas/casillas enumeradas, con primera casilla.
* Una **unidad de control** con un **cabezal lector/escritor**.



* En un instante determinado:
  - La cinta contiene sólo un número finito de símbolos distintos de **B**.
  - Cada celda contiene un símbolo del alfabeto de trabajo (el símbolo de la primera celda es $\triangleright$).
  - El cabezal inspecciona el contenido de una celda (**lee**), lo sustituye por un símbolo (**escribe**), y se desplaza a lo largo de la cinta (una celda a la izquierda, si $x = \mathbf{L}$, una a la derecha, si $x = \mathbf{R}$, o se queda en el mismo sitio, si $x = \mathbf{S}$)
  - La máquina se encuentra en un **estado** (elemento de Q).

Función de transición

¿Qué significa que la máquina M **ejecute** $\delta(q, s)$ en un cierto instante?

* Si $\delta(q, s) = (q', s', x)$ entonces la máquina M:
  * Pasa del estado **q** al estado **q'**.
  * Sustituye el símbolo **s** por el símbolo **s'**.
  * El cabezal se *desplaza* según el valor de $x \in \{$ **L**, **R**, **S** $\}$.
* La condición $\delta(q, \triangleright) = (q', s', x) \Rightarrow s' = \triangleright \wedge$ **x=R** significa lo siguiente:
  * si el cabezal está inspeccionando la primera celda, entonces el símbolo $\triangleright$ **no se modifica** y el cabezal se desplazará una celda a la derecha.

Máquina de Turing determinista: Semántica

Una **configuración** (**descripción instantánea**) de una MTD, M, en un instante t es una terna ordenada (**q**, **w**, **u**) tal que $q \in Q, w \in \Sigma^*$ y $u \in \Sigma^*$.

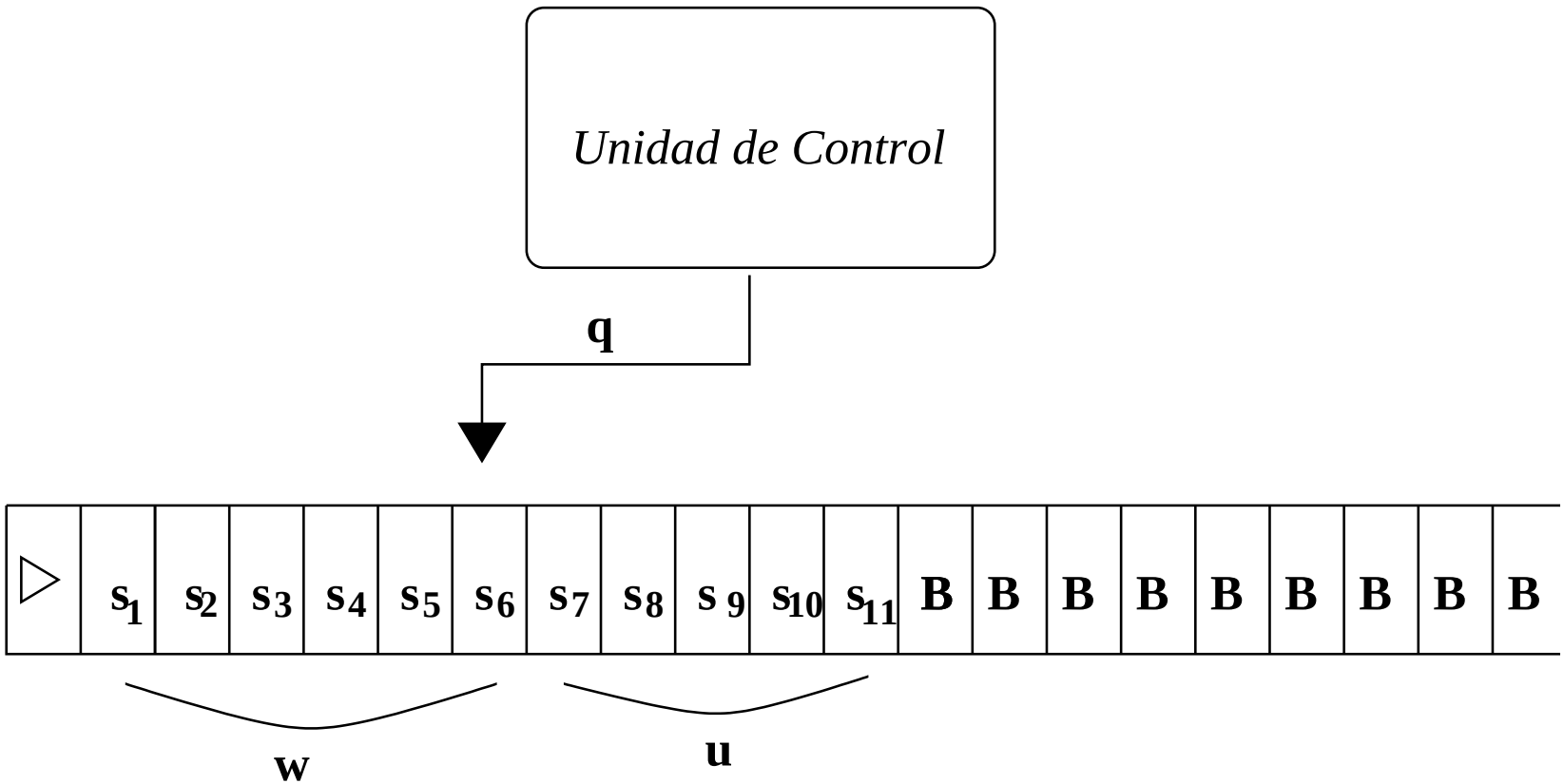
Una configuración (**q**, **w**, **u**) de una máquina MTD, M, en un instante t, **puede ser interpretada** como sigue:

- ⋆ **q** es el estado de la máquina M en el instante t.
- ⋆ ▷ **w u** es la palabra *escrita* en la cinta de la máquina M, en ese instante (el resto de símbolos de la cinta serán todos iguales a **B**).
- ⋆ Si **w** no es la cadena vacía, el cabezal de la máquina M estará **leyendo** el último símbolo de w, en ese instante. Caso contrario, estará leyendo el símbolo ▷ de la primera celda.

Genéricamente escribiremos: $\mathbf{C_t} = (\ \mathbf{q},\ \underbrace{w_1 \ldots w_{r-1}\ \overset{\downarrow}{w_r}}_{\mathbf{w}},\ \overbrace{u_1 u_2 \ldots u_s}^{\mathbf{u}}\ )$.

Máquina de Turing determinista: Semántica

Configuración C_t = (**q**, **w**, **u**) de una MTD en un instante t.



Máquina de Turing determinista: Semántica

Tipos interesantes de configuraciones.

Sea $\mathbf{C_t} = (\mathbf{q}, \mathbf{w}, \mathbf{u})$ una configuración de una MTD en un instante t:

* $\mathbf{C_t}$ es la **configuración inicial** asociada al dato de entrada u, si $q = q_0$, w es la cadena vacía λ y $u \in (\Sigma - \{B\})^*$. Notaremos $\mathbf{C_t} = \mathbf{I_u}$.

* $\mathbf{C_t}$ es una **configuración de parada**, si $\mathbf{q} \in F$.

* $\mathbf{C_t}$ es una **configuración de aceptación**, si $\mathbf{q} = q_y$.

* $\mathbf{C_t}$ es una **configuración de rechazo**, si $\mathbf{q} = q_n$.

Máquina de Turing determinista: Semántica

Sea $\mathbf{C_t} = (\ \mathbf{q},\ \underbrace{w_1 \ldots w_{r-1}\ \overset{\downarrow}{w_r}}_{\mathbf{w}},\ \overbrace{u_1 u_2 \ldots u_s}^{\mathbf{u}})$ una configuración de una MTD, en un instante t, tal que **no sea de parada**.

La configuración siguiente, $\mathbf{C_{t+1}}$, de $\mathbf{C_t}$ se obtiene aplicando la función de transición δ al par ordenado (q, w_r).

* **Caso 1:** $\boxed{\delta(q, w_r) = (q', w_r', L)}$

  En este caso, $\mathbf{C_{t+1}} = (\mathbf{q'}, \underbrace{w_1 \ldots \overset{\downarrow}{w_{r-1}}}, \overbrace{\mathbf{w_r'} u_1 u_2 \ldots u_s})$.

* **Caso 2:** $\boxed{\delta(q, w_r) = (q', w_r', R)}$

  En este caso, $\mathbf{C_{t+1}} = (\mathbf{q'}, \underbrace{w_1 \ldots w_{r-1} \mathbf{w_r'}\ \overset{\downarrow}{u_1}}, \overbrace{u_2 \ldots u_s})$.

* **Caso 3:** $\boxed{\delta(q, w_r) = (q', w_r', S)}$

  En este caso, $\mathbf{C_{t+1}} = (\mathbf{q'}, \underbrace{w_1 \ldots w_{r-1}\ \overset{\downarrow}{\mathbf{w_r'}}}, \overbrace{u_1 u_2 \ldots u_s})$.

Máquina de Turing determinista: Semántica

Computación $\mathcal{C}$ de una MTD, M: sucesión finita o infinita ($\mathbf{C_0}$, $\mathbf{C_1}, \ldots, \mathbf{C_r}$) de configuraciones tal que

* $\mathbf{C_0}$ es una configuración inicial (asociada a un dato de entrada $u \in (\Sigma - \{B\})^*$).
* Para cada $t < r$, la configuración $\mathbf{C_{t+1}}$ es la configuración siguiente de $\mathbf{C_t}$.

La **computación** $\mathcal{C}$ es **de parada** si

- $r \in \mathbf{N}$ y la última configuración $\mathbf{C_r}$, es (q, λ, u'), siendo $q \in \{q_y, q_n, q_h)$ y $u' \in (\Sigma - \{B\})^*$.
  - Si $q = q_y$, diremos que M **acepta** u y escribiremos **M(u)**=`yes`.
  - Si $q = q_n$, diremos que M **rechaza** u y escribiremos **M(u)**=`no`.
  - Si $q = q_h$, diremos que el **resultado** de la computación es la cadena $u' \in (\Sigma - \{B\})^*$ y escribiremos **M(u)= u'**.

Si la computación $\mathcal{C}$ es de parada, diremos que M **para sobre u** ($M(u)\downarrow$).

Si la computación $\mathcal{C}$ no es de parada, diremos que M **no para sobre u** ($M(u)\uparrow$).

MTD de decisión

Una **MTD** es de **decisión** si sus estados finales son $\mathbf{q_y}$ y $\mathbf{q_n}$.

Sea L un lenguaje sobre el alfabeto $\Sigma - \{B\}$.

Si M es una MTD de decisión, diremos que:

- $\star$ M **decide el lenguaje** L **sii** para cada $u \in (\Sigma - \{B\})^*$ se verifica:
  - Si $u \in L$, entonces **M(u)**=`yes`; es decir, si el estado de la configuracion de parada es $\mathbf{q_y}$.
  - Si $u \notin L$, entonces **M(u)**=`no`; es decir, si el estado de la configuracion de parada es $\mathbf{q_n}$.

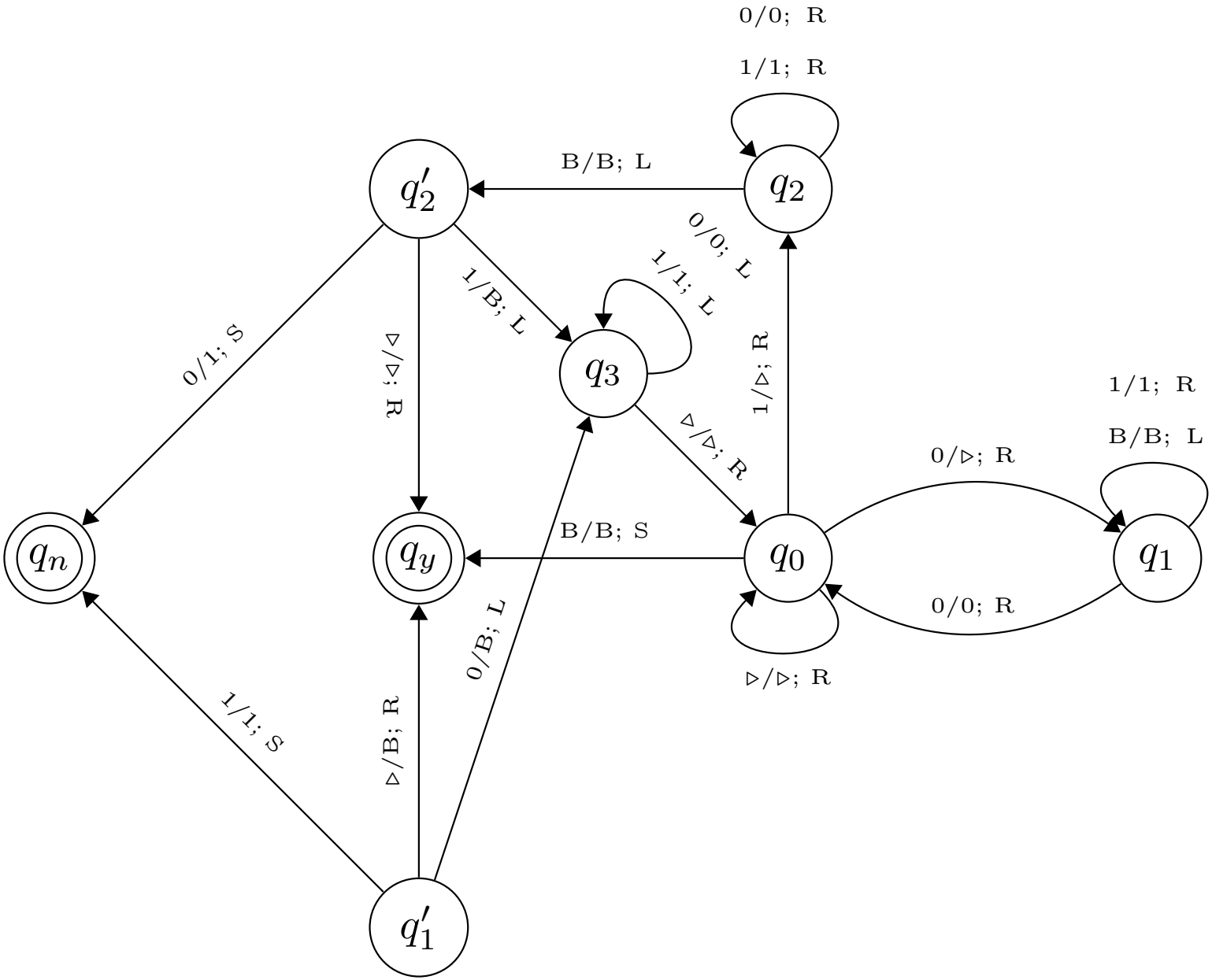
Ejemplo de una MTD de decisión

$Q = \{q_0, q_1, q_2, q_1', q_2', q_3, \mathbf{q_y}, \mathbf{q_n}\}$; $\Sigma = \{0, 1, B, \triangleright\}$

Función de transición:

$(Q - F) \times \Sigma$	$Q \times \Sigma \times \{L, R, S\}$
$(q_0, 0)$	$(q_1, \triangleright, R)$
$(q_0, 1)$	$(q_2, \triangleright, R)$
(q_0, B)	(q_y, B, S)
$(q_0, \triangleright)$	$(q_0, \triangleright, R)$
$(q_1, 0)$	$(q_0, 0, R)$
$(q_1, 1)$	$(q_1, 1, R)$
(q_1, B)	(q_1', B, L)
$(q_2, 0)$	$(q_2, 0, R)$
$(q_2, 1)$	$(q_2, 1, R)$
(q_2, B)	(q_2', B, L)
$(q_1', 0)$	(q_3, B, L)
$(q_1', 1)$	$(q_n, 1, S)$
$(q_1', \triangleright)$	(q_y, B, R)
$(q_2', 0)$	$(q_n, 1, S)$
$(q_2', 1)$	(q_3, B, R)
$(q_2', \triangleright)$	$(q_y, \triangleright, R)$
$(q_3, 0)$	$(q_3, 0, L)$
$(q_3, 1)$	$(q_3, 1, L)$
$(q_3, \triangleright)$	$(q_0, \triangleright, R)$

Diagrama de una MTD de decisión



MTD que calcula una función

Sea f una **función** parcial f de $(\Sigma - \{B\})^*$ en Σ^*

Diremos que una **MTD**, M, **calcula** la **función** f si se verifican las condiciones siguientes:

- El único estado final de M es $\mathbf{q_h}$.
- Para cada $u \in (\Sigma - \{B\})^*$ se tiene que:
  - La máquina M con dato de entrada u para **sii** $u \in \text{dom}(f)$.
  - Si la máquina M con dato de entrada u para entonces $M(u) = f(u)$.

Ejemplo de una MTD que calcula una función

Un ejemplo: $Q = \{q_0, q_1, \mathbf{q_h}\}$; $\Sigma = \{0, 1, B, \triangleright\}$

Función de transición:

$(Q - F) \times \Sigma$	$Q \times \Sigma \times \{L, R, S\}$
$(q_0, 0)$	$(q_0, 0, R)$
$(q_0, 1)$	$(q_0, 1, R)$
(q_0, B)	(q_1, B, L)
$(q_0, \triangleright)$	$(q_0, \triangleright, R)$
$(q_1, 0)$	$(q_h, 1, S)$
$(q_1, 1)$	$(q_1, 0, L)$
(q_1, B)	(q_1, B, S)
$(q_1, \triangleright)$	$(q_h, \triangleright, R)$

Hallemos $M(01)$

1. $\triangleright \overset{q_0}{0}\ 1\ B.$
2. $\triangleright\ \overset{q_0}{0}\ 1\ B.$
3. $\triangleright\ 0\ \overset{q_0}{1}\ B.$
4. $\triangleright\ 0\ 1\ \overset{q_0}{B}.$
5. $\triangleright\ 0\ \overset{q_1}{1}\ B.$
6. $\triangleright\ \overset{q_1}{0}\ 0\ B.$
7. $\triangleright\ \overset{q_h}{1}\ 0\ B.$

MTD's que resuelven problemas

¿Cómo **resolver un problema** X?

- Sea X un problema de decisión.
  - Una **MTD resuelve** X **sii decide** el lenguaje asociado a X.
- Sea X un problema caracterizado por una función total $f_X : L_X \rightarrow \Sigma^*$.
  - Una **MTD resuelve** X **sii calcula** la función f_X.

Máquina de Turing determinista con k cintas

- ⋆ En cada instante, la máquina está en **un estado**.
- ⋆ Consta de k cintas, infinitas a la derecha con primera celda/casilla.
- ⋆ Cada cinta tiene un cabezal lector/escritor que en cada instante:
  - Analiza una celda/casilla.
  - Puede reescribir sobre la celda/casilla.
  - Puede cambiar de estado (el *mismo* en todas las celdas).
  - Se puede desplazar (**L**, **R**, **S**) sobre la cinta.

Máquina de Turing determinista con k cintas

Para realizar una computación con entrada $u \in (\Sigma - \{B\})^*$:

- Se registra la entrada $u \in (\Sigma - \{B\})^*$ en la primera cinta.
- Las restantes cintas están en blanco.
- Inicialmente, todos los cabezales inspeccionan la primera celda.
- Inicialmente, la máquina está en el estado q_0.
- La función de transición **actúa**, simultáneamente, sobre cada cinta.

$$\delta(q, w^{(1)}, \ldots, w^{(k)}) = (q', w'^{(1)}, x^{(1)}, \ldots, w'^{(k)}, x^{(k)})$$

En donde $x^{(1)}, \ldots, x^{(k)} \in \{\mathbf{L}, \mathbf{R}, \mathbf{S}\}$.

Máquina de Turing determinista con k cintas

El resultado de una computación de parada es:

- `yes`, si el estado de la configuración de parada es q_y.
- `no`, si el estado de la configuración de parada es q_n.
- El contenido de la primera cinta, si el estado de la configuración de parada es q_h.

Teorema : Para cada MTD, M, *con k cintas* existe una MTD, M', con *una cinta* tal que $M(w) = M'(w)$, para cada dato de entrada w.

¿Qué **peaje se ha de pagar** para "pasar" de una MTD con k cintas a una MTD con una sóla cinta que sea "equivalente" a ella?

Máquina de Turing no determinista

Es una tupla, $M = (Q, \Sigma, q_0, F, \mathbf{B}, \triangleright, \delta)$, en donde:

- Q es un alfabeto finito (**estados**).
- Σ es un alfabeto finito (**de trabajo de la máquina**) tal que $Q \cap \Sigma = \emptyset$.
- $q_0 \in Q$ (**estado inicial**).
- $F = \{q_h, q_y, q_n\} \subseteq Q - \{q_0\}$ (**estados finales**, todos distintos entre sí).
- $B \in \Sigma$ (**símbolo blanco**).
- $\triangleright \in \Sigma$ (**primer símbolo**), $\triangleright \neq \mathbf{B}$.
- δ es una aplicación (**función de transición**) de $(Q - F) \times \Sigma$ en $\mathbf{P}(Q \times \Sigma \times \{\mathbf{L}, \mathbf{R}, \mathbf{S}\})$ tal que:
  - Si $(q', s', x) \in \delta(q, \triangleright)$ entonces $s' = \triangleright \wedge x = \mathbf{R}$.

MTD versus MTND

Es una tupla $M = (Q, \Sigma, q_0, F, \mathbf{B}, \triangleright, \delta)$:

Máquina de Turing determinista

- $\star$ Q es un conjunto finito no vacío (*estados*).
- $\star$ Σ es un alfabeto (*de la máquina*), $\Sigma \cap Q = \emptyset$.
- $\star$ $q_0 \in Q$ (*estado inicial*).
- $\star$ $F = \{q_h, q_y, q_n\} \subseteq Q - \{q_0\}$ (*estados finales*).
- $\star$ $\mathbf{B} \in \Sigma$ (*símbolo blanco*).
- $\star$ $\triangleright \in \Sigma$ (*primer símbolo*), $\triangleright \neq \mathbf{B}$.
- $\delta : (Q - F) \times \Sigma \to \mathbf{Q} \times \mathbf{\Sigma} \times \{\mathbf{L}, \mathbf{R}, \mathbf{S}\}$

Máquina de Turing no determinista

- $\star$ Q es un conjunto finito no vacío (*estados*).
- $\star$ Σ es un alfabeto (*de la máquina*), $Q \cap \Sigma = \emptyset$.
- $\star$ $q_0 \in Q$ (*estado inicial*).
- $\star$ $F = \{q_h, q_y, q_n\} \subseteq Q - \{q_0\}$ (*estados finales*).
- $\star$ $\mathbf{B} \in \Sigma$ (*símbolo blanco*).
- $\star$ $\triangleright \in \Sigma$ (*primer símbolo*), $\triangleright \neq \mathbf{B}$.
- $\delta : (Q - F) \times \Sigma \to \mathbf{P}(\mathbf{Q} \times \mathbf{\Sigma} \times \{\mathbf{L}, \mathbf{R}, \mathbf{S}\})$

Por ejemplo: en una MTD podría ser $\delta(q, s) = (q_1, s_1, \mathbf{L})$, mientras que en una MTND podría ser $\delta(q, s) = \{(q_1, s_1, \mathbf{S}), (q_2, s_2, \mathbf{R}), (q_3, s_3, \mathbf{L}), (q_4, s_4, \mathbf{R})\}$

Máquina de Turing no determinista

Para MTNDs, los conceptos de **configuración** en un instante t y **computacion** se definen de manera similar a como se hizo en las MTDs.

Observaciones importantes:

- Una configuración de una MTND que no sea de parada, **puede tener** más de una configuración siguiente.
- Dada una MTND, M, y un dato de entrada u pueden existir varias computaciones de M sobre u.
- Toda MTD es, en particular, una MTND.
- Una MTND de decisión es aquella cuyos estados finales son $\mathbf{q_y}$ y $\mathbf{q_n}$.

MTNDs que resuelven problemas de decisión

Sea L un lenguaje sobre el alfabeto $\Sigma - \{B\}$.

Si M es una MTND de decisión, diremos que:

- $\star$ M **decide** el lenguaje L **sii** para cada $u \in (\Sigma - \{B\})^*$ se verifica:
  - $u \in L$ **sii existe, al menos,** una computación de $M(u)$ tal que para y devuelve yes.

Teorema : Sea L un lenguaje tal que una MTND, M, decide L. Entonces existe otra MTD, M', con *tres cintas* que también decide L (ver **ANEXO**).

¿Qué **peaje se ha de pagar** para "pasar" de una MTND que decide L a una MTD que también decide L?

Una **MTND resuelve un problema de decisión** X **sii decide** el lenguaje asociado a X.

ANEXO

Teorema: *Toda* **MTND** *puede ser* simulada *por una* **MTD** *con* **tres cintas**.

Idea de la demostración:

Sea M una MTND (p: número máximo de elecciones no deterministas en M).

Para cada entrada u de M se considera el árbol de computación T_u asociado:

- ⋆ La raíz es la configuración inicial asociada a u.
- ⋆ Los nodos son las distintas configuraciones.
- ⋆ Existe un arco de C_1 a C_2 **sii** existe una transición en M de C_1 a C_2.

Se simulará el árbol de computación T_u mediante un *recorrido en anchura*.

- ⋆ En primer lugar, todas las configuraciones obtenidas al ejecutar **un** paso.
- ⋆ En primer lugar, todas las configuraciones obtenidas al ejecutar **dos** pasos.
- ⋆ Y así sucesivamente.

Una *dirección* es una cadena del alfabeto $\Gamma_p = \{1, \ldots, p\}$

- ⋆ Cada nodo de T_u tiene asociada unívocamente una dirección (*válidas*).
- ⋆ Existen direcciones que no se corresponden con ningún nodo (*no válidas*).

Las cadenas de Γ_P^* se ordenan lexicográficamente, según su longitud.

Descripción informal de una MTD, M', con **tres** cintas que *simula* a M:

- ⋆ Cinta 1: de entrada.
- ⋆ Cinta 2: de simulaciones.
- ⋆ Cinta 3: de direcciones.

Un esquema algorítmico que implementa una MTD, M', es el siguiente:

1. Inicialmente, la cinta 1 contiene a u y las cintas 2 y 3 están vacías
2. Colocar en la cinta 3 la primera cadena de Γ_P^*
3. Borrar la cinta 2, y copiar la cinta 1 en la cinta 2
4. Usar la cinta 2 para simular la parte de la computación *correspondiente* a la cadena de Γ_P^* que aparece en la cinta 3. Para ello:
   - ★ Se analiza el primer símbolo de la cadena de la cinta 3.
   - ★ (*) Si es *válido*, entonces
     - Realizar la correspondiente elección no determinista
     - Si ha llegado a una configuración de aceptación, entonces **aceptar** y **terminar**
     - Si no,

       si queda algún símbolo por analizar en la cinta 3, elegir el siguiente e ir a (*)

       si no, ir al paso 5

     si no es *válido*, ir al paso 5
5. Reemplazar la cadena de la cinta 3 por la *siguiente* cadena.
   - ★ Si no hay siguiente cadena, entonces **terminar**
6. Ir al paso 3.