

Teoría de la Complejidad Computacional

Tema 2: Modelos de Computación Máquinas de Turing

Mario de J. Pérez Jiménez

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

marper@us.es

<http://www.cs.us.es/~marper/>

Máster Universitario en Matemáticas
Curso 2019-2020



Índice

- * Modelos de computación.
- * Máquinas de Turing deterministas.
 - Sintaxis.
 - 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.

Modelos de computación

Formalización del concepto de procedimiento mecánico.

- ★ **Sintaxis.**
- ★ **Semántica.**

En un modelo de computación se tiene:

- ★ **Procedimiento mecánico.**
- ★ **Funciones computables.**
- ★ **Máquinas.**

De ahí que existan modelos de computación:

- Orientados a **programas** (Modelo GOTO, modelo WHILE, etc.)
- Orientados a **funciones** (Funciones recursivas, λ -calculables, etc.)
- Orientados a **máquinas** (Máquinas de Turing, de Post, URM, etc.)

Máquina de Turing determinista: Sintaxis

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

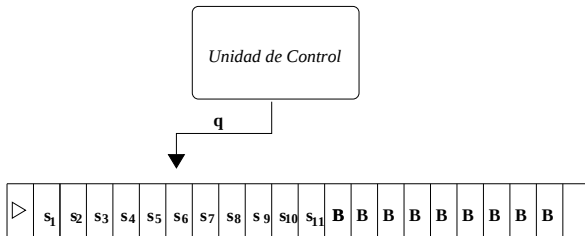
- Q es un conjunto finito no vacío (**estados**).
- Σ es un alfabeto (**de trabajo de la máquina**) tal que $\Sigma \cap Q = \emptyset$.
- δ es una aplicación (**función de transición**) de $(Q - F) \times \Sigma$ en $Q \times \Sigma \times \{0, 1, -1\}$ tal que:

★ Si $\delta(q, \triangleright) = (q', s', x)$ entonces $s' = \triangleright \wedge x = 1$.

- $q_0 \in Q$ (**estado inicial**).
- $F = \{q_h, q_y, q_n\} \subseteq Q - \{q_0\}$ (**estados finales**).
- $B \in \Sigma$ (**símbolo blanco**).
- $\triangleright \in \Sigma$ (**primer símbolo**) verificando que $\triangleright \neq B$.

Informalmente, una máquina de Turing determinista M consta de:

- ★ Una **cinta infinita** estructurada en casillas enumeradas (con primera casilla), cada una conteniendo un símbolo del alfabeto de trabajo.
- ★ Un **cabezal** de trabajo **lector/escritor** que inspecciona el contenido de una casilla (**lee**), puede cambiar su contenido por otro símbolo (**escribe**) y se desplaza a lo largo de la cinta (una casilla a la derecha, una casilla a la izquierda o se queda en el mismo sitio).



Si $\delta(q, s) = (q', s', x)$, diremos que M pasa de q a q' , sustituye s por s' y se *desplaza* según el valor de $x \in \{0, 1, -1\}$.

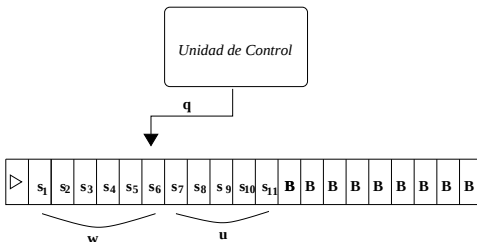
Máquina de Turing determinista: **Semántica**

Configuración (*descripción instantánea*) de una máquina de Turing M :

- * Una terna (q, w, u) tal que $q \in Q$, $w \in \Sigma^*$ y $u \in \Sigma^*$.

Una configuración (q, w, u) **puede ser interpretada** como sigue:

- ★ q es el estado actual en que se encuentra la máquina M .
- ★ $\triangleright w u$ es la palabra *escrita* en la cinta de la máquina M .
- ★ Si w no es la cadena vacía, la máquina M *analiza* el último símbolo de w .
Caso contrario, analiza el símbolo $\triangleright$ de la primera casilla.



Máquina de Turing determinista: Semántica

Sea $C = (q, w, u)$ una configuración de M .

- * Si $q = q_0$, w es la cadena vacía λ y $u \in (\Sigma - \{B\})^*$: C es la **configuración inicial** con dato de entrada u y notaremos $C = I_u$.
- * Si $q \in F$: C es una **configuración de parada**.
- * Si $q = q_y$: C es una **configuración de aceptación**.
- * Si $q = q_n$: C es una **configuración de rechazo**.

Configuración siguiente: Sea $(q, w_1 \dots w_{r-1} w_r, u_1 u_2 \dots u_s)$ una configuración de M .

- * Si $\delta(q, w_r) = (q', w'_r, -1)$, diremos que la configuración siguiente de $(q, \underbrace{w_1 \dots w_{r-1}}_{\downarrow}, \underbrace{u_1 u_2 \dots u_s}_{\downarrow})$ es $(q', \underbrace{w_1 \dots w_{r-1}}_{\downarrow}, \underbrace{w'_r u_1 u_2 \dots u_s}_{\downarrow})$.
- * Si $\delta(q, w_r) = (q', w'_r, +1)$, diremos que la configuración siguiente de $(q, \underbrace{w_1 \dots w_{r-1}}_{\downarrow}, \underbrace{u_1 u_2 \dots u_s}_{\downarrow})$ es $(q', \underbrace{w_1 \dots w_{r-1} w'_r}_{\downarrow}, \underbrace{u_1 u_2 \dots u_s}_{\downarrow})$.
- * Si $\delta(q, w_r) = (q', w'_r, 0)$, diremos que la configuración siguiente de $(q, \underbrace{w_1 \dots w_{r-1}}_{\downarrow}, \underbrace{u_1 u_2 \dots u_s}_{\downarrow})$ es $(q, \underbrace{w_1 \dots w_{r-1} w'_r}_{\downarrow}, \underbrace{u_1 u_2 \dots u_s}_{\downarrow})$.

Máquina de Turing determinista: Semántica

Una **computación** $\mathcal{C}$ de una MT es una sucesión (finita o infinita) de configuraciones $(C_0, C_1, \dots, C_r)$, $r \in \mathbf{N} \cup \{\infty\}$, tal que

- * C_0 es una configuración inicial (asociada a un dato de entrada $u \in (\Sigma - \{B\})^*$).
- * Para cada $i < r$, la configuración C_{i+1} es la configuración siguiente de C_i .
- * La computación $\mathcal{C}$ es de parada si

- $r \in \mathbf{N}$ y la última configuración 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** el dato u y escribiremos $M(u) = \text{yes}$.

- ★ Si $q = q_n$, diremos que M **rechaza** el dato u y escribiremos $M(u) = \text{no}$.

- ★ Si $q = q_h$, diremos que el **resultado** de la computación es $u' \in (\Sigma - \{B\})^*$ y escribiremos $M(u) = u'$.

- * Si la computación $\mathcal{C}$ es de parada, escribiremos $M(u) \downarrow$
- * Si la computación $\mathcal{C}$ no es de parada, escribiremos $M(u) \uparrow$

Decisión en una MTD

Una **MTD** es **de decisión** si sus estados finales son q_y y q_n .

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

Si M es una MTD de decisión, diremos que:

★ M **decide el lenguaje L** si y solo si para cada $u \in (\Sigma - \{B\})^*$ se verifica:

- Si $u \in L$, entonces $M(u) = \text{yes}$ (el estado de la configuración de parada es q_y).
- Si $u \notin L$, entonces $M(u) = \text{no}$ (el estado de la configuración de parada es q_n).

Ejemplo de una MTD de decisión

$$Q = \{q_0, q_1, q_2, q'_1, q'_2, q_3, q_h, q_y, q_n\}; \Sigma = \{0, 1, B, \triangleright\}$$

$(Q - F) \times \Sigma$	$Q \times \Sigma \times \{0, 1, -1\}$
$(q_0, 0)$	$(q_1, \triangleright, 1)$
$(q_0, 1)$	$(q_2, \triangleright, 1)$
(q_0, B)	$(q_y, B, 0)$
$(q_0, \triangleright)$	$(q_0, \triangleright, 1)$
$(q_1, 0)$	$(q_1, 0, 1)$
$(q_1, 1)$	$(q_1, 1, 1)$
(q_1, B)	$(q'_1, B, -1)$
$(q_2, 0)$	$(q_2, 0, 1)$
$(q_2, 1)$	$(q_2, 1, 1)$
(q_2, B)	$(q'_2, B, -1)$
$(q'_1, 0)$	$(q_3, B, -1)$
$(q'_1, 1)$	$(q_n, 1, 0)$
$(q'_1, \triangleright)$	$(q_y, B, 1)$
$(q'_2, 0)$	$(q_n, 1, 0)$
$(q'_2, 1)$	$(q_3, B, -1)$
$(q'_2, \triangleright)$	$(q_y, \triangleright, 1)$
$(q_3, 0)$	$(q_3, 0, -1)$
$(q_3, 1)$	$(q_3, 1, -1)$
$(q_3, \triangleright)$	$(q_0, \triangleright, 1)$

MTD's que calculan funciones

- ★ Una MTD, M , **calcula** una función parcial $f : (\Sigma - \{B\})^* \rightarrow \Sigma^*$ si y solo si para cada $u \in (\Sigma - \{B\})^*$ se tiene que:

- * $M \downarrow u$ si y solo si $f(u) \downarrow$.
- * Si $M \downarrow u$ entonces $M(u) = f(u)$.

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

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

Hallemos $M(01)$

1. $\overset{q_0}{\triangleright} 0 1 B.$

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** si y solo si 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** si y solo si 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 casilla.
- ★ Cada cinta tiene un cabezal lector/escritor que en cada instante:
 - Analiza una casilla.
 - Puede reescribir sobre la casilla.
 - Puede cambiar de estado (el mismo en todas las casillas).
 - Se puede desplazar: $+1, -1, 0$.

- ★ Para realizar una computación con entrada $u \in \Sigma^*$:
 - Se registra la entrada $u \in \Sigma^*$ en la primera cinta.
 - Las restantes cintas están en blanco.
 - Todos los cabezales apuntan a la primera casilla.
 - La máquina está en el estado inicial.
 - La función de transición *actúa* sobre cada cinta.

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 cinta k -ésima, 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 entrada w .

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

Máquina de Turing no determinista

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

- Q es un conjunto finito no vacío (**estados**).
- Σ es un alfabeto (**de trabajo de la máquina**), $Q \cap \Sigma = \emptyset$.
- $\delta : (Q - F) \times \Sigma \rightarrow \mathbf{P}(Q \times \Sigma \times \{0, 1, -1\})$ (**función de transición**).
- $q_0 \in Q$ (**estado inicial**).
- $F = \{q_h, q_y, q_n\} \subseteq Q - \{q_0\}$ (**estados finales**).
- $B \in \Sigma$ (**símbolo blanco**)
- $\triangleright \in \Sigma$ (**primer símbolo**), $\triangleright \neq B$.

MTD versus MTND

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

Máquina de Turing determinista

- ★ Q es un conjunto finito no vacío (*estados*).
- ★ Σ es un alfabeto (*de la máquina*), $\Sigma \cap Q = \emptyset$.
- $\delta : (Q - F) \times \Sigma \rightarrow Q \times \Sigma \times \{0, 1, -1\}$
- ★ $q_0 \in Q$ (*estado inicial*).
- ★ $F = \{q_h, q_y, q_n\} \subseteq Q - \{q_0\}$ (*estados finales*).
- ★ $B \in \Sigma$ (*símbolo blanco*).
- ★ $\triangleright \in \Sigma$ (*primer símbolo*), $\triangleright \neq B$.

Máquina de Turing no determinista

- ★ Q es un conjunto finito no vacío (*estados*).
- ★ Σ es un alfabeto (*de la máquina*), $Q \cap \Sigma = \emptyset$.
- $\delta : (Q - F) \times \Sigma \rightarrow P(Q \times \Sigma \times \{0, 1, -1\})$
- ★ $q_0 \in Q$ (*estado inicial*).
- ★ $F = \{q_h, q_y, q_n\} \subseteq Q - \{q_0\}$ (*estados finales*).
- ★ $B \in \Sigma$ (*símbolo blanco*).
- ★ $\triangleright \in \Sigma$ (*primer símbolo*), $\triangleright \neq B$.

Máquina de Turing no determinista

Los conceptos de d.i. y computacion en MTNDs se definen de manera similar a como se hizo con las MTDs.

- Una d.i. de una MTND puede tener más de una d.i. siguiente.
- Dada una MTND, M , y un dato de entrada u pueden existir muchas computaciones de M sobre u .
- Toda MTD es, en particular, una MTND.
- Una MTND de decisión es aquella cuyos estados finales son $\{q_y, q_n\}$.

Decisión en una MTND

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

Si M es una MTND de decisión, diremos que:

- ★ M **decide** el lenguaje L si y solo si para cada $u \in (\Sigma - \{B\})^*$ se verifica:
 - $u \in L$ si y sólo si 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 si y solo si 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 si y solo si 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 $\Gamma_p = \{1, \dots, p\}$

- ★ 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 algoritmo 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.