

Máquinas de Turing

IIC3242

Complejidad Computacional

Objetivo: Medir la **complejidad computacional** de un problema.

Vale decir: Medir la cantidad de **recursos computacionales** necesarios para solucionar un problema.

- ▶ Tiempo
- ▶ Espacio
- ▶ ...

Para hacer esto primero tenemos que introducir la noción de **problema**.

Problemas de decisión

Alfabeto Σ : Conjunto finito de símbolos.

- ▶ Ejemplo: $\Sigma = \{0, 1\}$

Palabra w : Secuencia finita de símbolos de Σ .

- ▶ Ejemplo: $w = 01101$

Σ^* : Conjunto de todas las palabras construidas con símbolos de Σ .

Lenguaje L : Conjunto de palabras.

- ▶ Ejemplo: $L = \{0^n 1^n \mid n \in \mathbb{N}\}$

Problemas de decisión

Problema de decisión asociado a un lenguaje L : Dado $w \in \Sigma^*$, decidir si $w \in L$.

Ejemplo

Podemos ver SAT como un problema de decisión. Suponga que $P = \{p, q\}$:

- ▶ $\Sigma = \{p, q, \neg, \wedge, \vee, \rightarrow, \leftrightarrow, (,)\}$

Algunas palabras de Σ^* representan fórmulas, mientras que otras tales como $\neg\neg$ y $p\neg q \wedge \wedge \vee q$ no representan fórmulas

- ▶ $\text{SAT} = \{w \in \Sigma^* \mid w \text{ representa una fórmula y } w \text{ es satisfacible}\}$

Complejidad de un problema de decisión

La complejidad de un lenguaje L es la complejidad del problema de decisión asociado a L .

¿Cuándo decimos que L puede ser solucionado eficientemente?

- ▶ Cuando existe un **algoritmo eficiente** que decide L

Ejercicio

Muestre que $L = \{w \in \{0,1\}^* \mid w \text{ es un palíndromo}\}$ puede ser resuelto eficientemente.

¿Cuándo decimos que L es un problema difícil?

- ▶ Cuando **no existe** un algoritmo eficiente que decide L

¿Cómo podemos demostrar que un problema es difícil?

- ▶ Para hacer esto, primero tenemos que formalizar la noción de algoritmo

¿Qué es un algoritmo? ¿Podemos formalizar este concepto?

- ▶ **Máquinas de Turing:** Intento por formalizar este concepto

¿Podemos demostrar que las máquinas de Turing capturan la noción de algoritmo?

- ▶ No, el concepto de algoritmo es intuitivo

Máquinas de Turing

¿Por qué creemos que las máquinas de Turing son una buena formalización del concepto de algoritmo?

Máquinas de Turing

¿Por qué creemos que las máquinas de Turing son una buena formalización del concepto de algoritmo?

- ▶ Porque cada **programa** de una máquina de Turing puede ser implementado

Máquinas de Turing

¿Por qué creemos que las máquinas de Turing son una buena formalización del concepto de algoritmo?

- ▶ Porque cada **programa** de una máquina de Turing puede ser implementado
- ▶ Porque todos los algoritmos conocidos han podido ser implementados en máquinas de Turing

Máquinas de Turing

¿Por qué creemos que las máquinas de Turing son una buena formalización del concepto de algoritmo?

- ▶ Porque cada **programa** de una máquina de Turing puede ser implementado
- ▶ Porque todos los algoritmos conocidos han podido ser implementados en máquinas de Turing
- ▶ Porque todos los otros intentos por formalizar este concepto fueron reducidos a las máquinas de Turing

Máquinas de Turing

¿Por qué creemos que las máquinas de Turing son una buena formalización del concepto de algoritmo?

- ▶ Porque cada **programa** de una máquina de Turing puede ser implementado
- ▶ Porque todos los algoritmos conocidos han podido ser implementados en máquinas de Turing
- ▶ Porque todos los otros intentos por formalizar este concepto fueron reducidos a las máquinas de Turing
 - ▶ Los mejores intentos resultaron ser equivalentes a las máquinas de Turing
 - ▶ Todos los intentos “razonables” fueron reducidos **eficientemente**

Máquinas de Turing

¿Por qué creemos que las máquinas de Turing son una buena formalización del concepto de algoritmo?

- ▶ Porque cada **programa** de una máquina de Turing puede ser implementado
- ▶ Porque todos los algoritmos conocidos han podido ser implementados en máquinas de Turing
- ▶ Porque todos los otros intentos por formalizar este concepto fueron reducidos a las máquinas de Turing
 - ▶ Los mejores intentos resultaron ser equivalentes a las máquinas de Turing
 - ▶ Todos los intentos “razonables” fueron reducidos **eficientemente**
- ▶ Tesis de Church: **Algoritmo = Máquina de Turing**

Definición

Máquina de Turing (determinista): $(Q, \Sigma, \Gamma, q_0, \delta, F)$

- ▶ Q es un conjunto finito de estados
- ▶ Σ es un alfabeto tal que $\vdash, \sqcup \notin \Sigma$
- ▶ Γ es un alfabeto tal que $\Sigma \cup \{\vdash, \sqcup\} \subseteq \Gamma$
- ▶ $q_0 \in Q$ es el estado inicial
- ▶ $F \subseteq Q$ es un conjunto de estados finales
- ▶ δ es una función parcial:

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{\leftarrow, \sqcup, \rightarrow\}$$

δ es llamada función de transición

Máquinas de Turing: Funcionamiento

La cinta de la máquina de Turing es infinita hacia la derecha.

- ▶ El símbolo $\vdash$ es usado para demarcar la posición 0 de la cinta

Supuestos

- ▶ Si $\delta(q, \vdash)$ está definido: $\delta(q, \vdash) = (q', \vdash, X)$, con $X \in \{\rightarrow, \square\}$
- ▶ Si $a \in (\Gamma \setminus \{\vdash\})$ y $\delta(q, a)$ está definido: $\delta(q, a) = (q', b, X)$, con $b \in (\Gamma \setminus \{\vdash\})$

Σ es el alfabeto de entrada y Γ es el alfabeto de la cinta.

- ▶ Una palabra $w \in \Sigma^*$ de entrada de largo n es colocada en las posiciones $1, \dots, n$ de la cinta
- ▶ Las posiciones siguientes $(n + 1, n + 2, \dots)$ contienen el símbolo B

Máquinas de Turing: Funcionamiento

Al comenzar a funcionar, la máquina se encuentra en el estado q_0 y su cabeza lectora está en la posición 1 de la cinta.

En cada instante la máquina se encuentra en un estado q y su cabeza lectora está en una posición p .

- ▶ Si el símbolo en la posición p es a y $\delta(q, a) = (q', b, X)$, entonces:

Máquinas de Turing: Funcionamiento

Al comenzar a funcionar, la máquina se encuentra en el estado q_0 y su cabeza lectora está en la posición 1 de la cinta.

En cada instante la máquina se encuentra en un estado q y su cabeza lectora está en una posición p .

- ▶ Si el símbolo en la posición p es a y $\delta(q, a) = (q', b, X)$, entonces:
 - ▶ La máquina escribe el símbolo b en la posición p de la cinta

Máquinas de Turing: Funcionamiento

Al comenzar a funcionar, la máquina se encuentra en el estado q_0 y su cabeza lectora está en la posición 1 de la cinta.

En cada instante la máquina se encuentra en un estado q y su cabeza lectora está en una posición p .

- ▶ Si el símbolo en la posición p es a y $\delta(q, a) = (q', b, X)$, entonces:
 - ▶ La máquina escribe el símbolo b en la posición p de la cinta
 - ▶ Cambia de estado desde q a q'

Máquinas de Turing: Funcionamiento

Al comenzar a funcionar, la máquina se encuentra en el estado q_0 y su cabeza lectora está en la posición 1 de la cinta.

En cada instante la máquina se encuentra en un estado q y su cabeza lectora está en una posición p .

- ▶ Si el símbolo en la posición p es a y $\delta(q, a) = (q', b, X)$, entonces:
 - ▶ La máquina escribe el símbolo b en la posición p de la cinta
 - ▶ Cambia de estado desde q a q'
 - ▶ Mueve la cabeza lectora a la posición $p - 1$ si X es $\leftarrow$, y a la posición $p + 1$ si X es $\rightarrow$. Si X es $\square$, entonces la cabeza lectora permanece en la posición p

Máquinas de Turing: Aceptación

Los estados de F son utilizados como estados de aceptación.

- Una palabra w es aceptada por una máquina M si y sólo si la ejecución de M con entrada w se detiene en un estado de F

Definición

Lenguaje aceptado por una máquina de Turing M :

$$L(M) = \{w \in \Sigma^* \mid M \text{ acepta } w\}$$

Máquinas de Turing: Ejemplo

Queremos construir una máquina que verifique si el número de 0s en una palabra es par: $M = (Q, \Sigma, \Gamma, q_0, \delta, F)$

- ▶ $Q = \{q_0, q_1\}$
- ▶ $\Sigma = \{0, 1\}$
- ▶ $\Gamma = \{0, 1, \vdash, B\}$
- ▶ $F = \{q_0\}$
- ▶ δ es definida como:

$$\delta(q_0, 0) = (q_1, B, \rightarrow)$$

$$\delta(q_0, 1) = (q_0, B, \rightarrow)$$

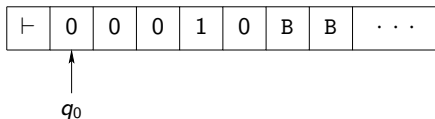
$$\delta(q_1, 0) = (q_0, B, \rightarrow)$$

$$\delta(q_1, 1) = (q_1, B, \rightarrow)$$

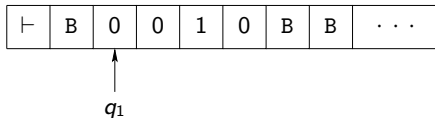
Máquinas de Turing: Ejecución

Supongamos que $w = 00010$:

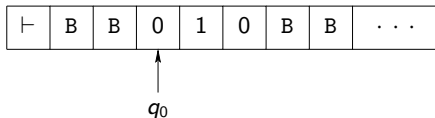
Inicio:



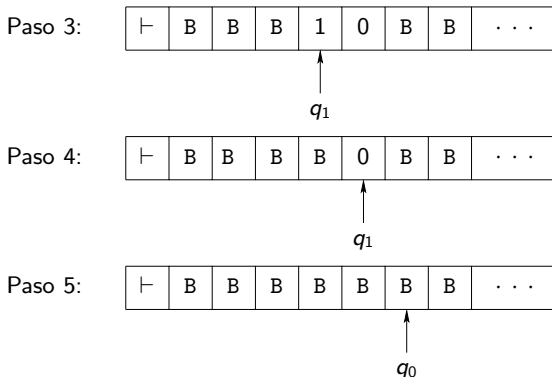
Paso 1:



Paso 2:



Máquinas de Turing: Ejecución



Conclusión: La máquina acepta $w = 00010$

El lenguaje aceptado por una máquina de Turing: Ejemplos

Ejemplo

Para la máquina M mostrada en las transparencias anteriores:

$$L(M) = \{w \in \{0, 1\}^* \mid w \text{ contiene un número par de símbolos } 0\}$$

El lenguaje aceptado por una máquina de Turing: Ejemplos

Ejemplo

Para la máquina M mostrada en las transparencias anteriores:

$$L(M) = \{w \in \{0,1\}^* \mid w \text{ contiene un número par de símbolos } 0\}$$

Ejercicio

Construya una máquina de Turing que acepte el lenguaje

$$L = \{w \in \{0,1\}^* \mid w \text{ es un palíndromo}\}.$$

Complejidad de un algoritmo

Una Máquina de Turing puede no detenerse en alguna entrada.

- ▶ **Primera noción de algoritmo:** MT que se detiene en todas las entradas

¿Cómo se mide el tiempo de ejecución de un algoritmo?

Para una MT con alfabeto Σ :

- ▶ **Paso de M :** Ejecutar una instrucción de la función de transición
- ▶ **$tiempo_M(w)$:** Número de pasos ejecutados por M con entrada $w \in \Sigma^*$

Definición

El tiempo de funcionamiento de una MT M en el *peor caso* es definido por la función t_M :

$$t_M(n) = \text{máx}\{ \text{tiempo}_M(w) \mid w \in \Sigma^* \text{ y } |w| = n \}.$$

Ejercicio

Construya una máquina de Turing que funcione en tiempo $O(n^2)$ y acepte el lenguaje $L = \{w \in \{0,1\}^* \mid w \text{ es un palíndromo}\}$

Definición

MT (determinista) con k cintas: $(Q, \Sigma, \Gamma, q_0, \delta, F)$

- ▶ Q es un conjunto finito de estados
- ▶ Σ es un alfabeto tal que $\vdash, \sqcup \notin \Sigma$
- ▶ Γ es un alfabeto tal que $\Sigma \cup \{\vdash, \sqcup\} \subseteq \Gamma$
- ▶ $q_0 \in Q$ es el estado inicial
- ▶ $F \subseteq Q$ es un conjunto de estados finales
- ▶ δ es una función parcial:

$$\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{\leftarrow, \sqcup, \rightarrow\}^k.$$

δ es llamada función de transición.

Máquinas de Turing con k cintas: Funcionamiento

La máquina tiene k cintas infinitas hacia la derecha.

- ▶ El símbolo $\vdash$ es usado para demarcar la posición 0 de cada cinta

Σ es el alfabeto de entrada y Γ es el alfabeto de las cintas.

- ▶ Una palabra $w \in \Sigma^*$ de entrada de largo n es colocada en las posiciones $1, \dots, n$ de la primera cinta
- ▶ Las siguientes posiciones $(n + 1, n + 2, \dots)$ de la primera cinta contienen el símbolo B
- ▶ Las restantes cintas contienen el símbolo B en las posiciones $1, 2, 3, \dots$

Máquinas de Turing con k cintas: Funcionamiento

La máquina tiene una cabeza lectora por cinta.

- ▶ Al comenzar, la máquina se encuentra en el estado q_0 , y cada cabeza lectora está en la posición 1 de su cinta

En cada instante la máquina se encuentra en un estado q y su cabeza lectora i se encuentra en la posición p_i .

- ▶ Si el símbolo en la posición p_i es a_i y $\delta(q, a_1, \dots, a_k) = (q', b_1, \dots, b_k, X_1, \dots, X_k)$, entonces:

Máquinas de Turing con k cintas: Funcionamiento

La máquina tiene una cabeza lectora por cinta.

- ▶ Al comenzar, la máquina se encuentra en el estado q_0 , y cada cabeza lectora está en la posición 1 de su cinta

En cada instante la máquina se encuentra en un estado q y su cabeza lectora i se encuentra en la posición p_i .

- ▶ Si el símbolo en la posición p_i es a_i y $\delta(q, a_1, \dots, a_k) = (q', b_1, \dots, b_k, X_1, \dots, X_k)$, entonces:
 - ▶ La máquina escribe el símbolo b_i en la posición p_i de la i -ésima cinta

Máquinas de Turing con k cintas: Funcionamiento

La máquina tiene una cabeza lectora por cinta.

- ▶ Al comenzar, la máquina se encuentra en el estado q_0 , y cada cabeza lectora está en la posición 1 de su cinta

En cada instante la máquina se encuentra en un estado q y su cabeza lectora i se encuentra en la posición p_i .

- ▶ Si el símbolo en la posición p_i es a_i y $\delta(q, a_1, \dots, a_k) = (q', b_1, \dots, b_k, X_1, \dots, X_k)$, entonces:
 - ▶ La máquina escribe el símbolo b_i en la posición p_i de la i -ésima cinta
 - ▶ Cambia de estado desde q a q'

Máquinas de Turing con k cintas: Funcionamiento

La máquina tiene una cabeza lectora por cinta.

- ▶ Al comenzar, la máquina se encuentra en el estado q_0 , y cada cabeza lectora está en la posición 1 de su cinta

En cada instante la máquina se encuentra en un estado q y su cabeza lectora i se encuentra en la posición p_i .

- ▶ Si el símbolo en la posición p_i es a_i y $\delta(q, a_1, \dots, a_k) = (q', b_1, \dots, b_k, X_1, \dots, X_k)$, entonces:
 - ▶ La máquina escribe el símbolo b_i en la posición p_i de la i -ésima cinta
 - ▶ Cambia de estado desde q a q'
 - ▶ Mueve la cabeza lectora de la i -ésima cinta a la posición $p_i - 1$ si X_i es $\leftarrow$, y a la posición $p_i + 1$ si X_i es $\rightarrow$. Si X_i es $\square$, entonces la máquina no mueve la cabeza de la i -ésima cinta

MT con k cintas: Aceptación y complejidad

Una palabra w es aceptada por una MT M con k cintas si y sólo si la ejecución de M con entrada w se detiene en un estado final.

$$L(M) = \{w \in \Sigma^* \mid M \text{ acepta } w\}$$

Para una MT con k cintas y alfabeto Σ :

- ▶ **Paso de M :** Ejecutar una instrucción de la función de transición
- ▶ **$tiempo_M(w)$:** Número de pasos ejecutados por M con entrada $w \in \Sigma^*$
- ▶ **Tiempo de funcionamiento M en el peor caso:**

$$t_M(n) = \max\{ tiempo_M(w) \mid w \in \Sigma^* \text{ y } |w| = n \}$$

MT con k cintas: Ejemplo

Ejercicio

Construya una MT M con dos cintas que funcione en tiempo $O(n)$ y acepte el lenguaje $L = \{w \in \{0,1\}^* \mid w \text{ es un palíndromo}\}$.

MT con k cintas: Ejemplo

Ejercicio

Construya una MT M con dos cintas que funcione en tiempo $O(n)$ y acepte el lenguaje $L = \{w \in \{0,1\}^* \mid w \text{ es un palíndromo}\}$.

Solución: Definimos $M = (Q, \Sigma, \Gamma, q_0, \delta, F)$ de la siguiente forma:

- ▶ $Q = \{q_0, q_c, q_r, q_v, q_a\}$
- ▶ $\Sigma = \{0, 1\}$
- ▶ $\Gamma = \{0, 1, B, \vdash\}$
- ▶ $F = \{q_a\}$

MT con k cintas: Ejemplo

- Función δ es definida de la siguiente forma:

$(q_0, B, B) \rightarrow (q_a, B, B, \square, \square)$	$(q_r, 1, 0) \rightarrow (q_r, 1, 0, \leftarrow, \square)$
$(q_0, 0, B) \rightarrow (q_c, 0, 0, \rightarrow, \rightarrow)$	$(q_r, 1, 1) \rightarrow (q_r, 1, 1, \leftarrow, \square)$
$(q_0, 1, B) \rightarrow (q_c, 1, 1, \rightarrow, \rightarrow)$	$(q_r, \vdash, 0) \rightarrow (q_v, \vdash, 0, \rightarrow, \square)$
$(q_c, 0, B) \rightarrow (q_c, 0, 0, \rightarrow, \rightarrow)$	$(q_r, \vdash, 1) \rightarrow (q_v, \vdash, 1, \rightarrow, \square)$
$(q_c, 1, B) \rightarrow (q_c, 1, 1, \rightarrow, \rightarrow)$	$(q_v, 0, 0) \rightarrow (q_v, 0, 0, \rightarrow, \leftarrow)$
$(q_c, B, B) \rightarrow (q_r, B, B, \leftarrow, \leftarrow)$	$(q_v, 1, 1) \rightarrow (q_v, 1, 1, \rightarrow, \leftarrow)$
$(q_r, 0, 0) \rightarrow (q_r, 0, 0, \leftarrow, \square)$	$(q_v, B, \vdash) \rightarrow (q_a, B, \vdash, \square, \square)$
$(q_r, 0, 1) \rightarrow (q_r, 0, 1, \leftarrow, \square)$	

Aceptación en distintos modelos

Un lenguaje L es aceptado por una MT M si $L = L(M)$.

- ▶ ¿Es posible aceptar más lenguajes si se usa cintas adicionales?

Aceptación en distintos modelos

Un lenguaje L es aceptado por una MT M si $L = L(M)$.

- ¿Es posible aceptar más lenguajes si se usa cintas adicionales?

Teorema

Si un lenguaje L es aceptado por una MT M_1 con k cintas, entonces L es aceptado por una MT M_2 con una cinta.

Aceptación en distintos modelos

Un lenguaje L es aceptado por una MT M si $L = L(M)$.

- ¿Es posible aceptar más lenguajes si se usa cintas adicionales?

Teorema

Si un lenguaje L es aceptado por una MT M_1 con k cintas, entonces L es aceptado por una MT M_2 con una cinta.

Ejercicio

Demuestre el teorema.

- ¿Cuál es la diferencia de complejidad entre M_1 y M_2 ?

Complejidad en distintos modelos

Un lenguaje L es aceptado por una MT M en tiempo $O(t(n))$ si $L = L(M)$ y $t_M(n)$ es $O(t(n))$.

- ▶ La definición es idéntica para el caso de $\Omega(t(n))$ y $\Theta(t(n))$

Esta definición considera a una Máquina de Turing con k cintas, para $k \geq 1$.

Complejidad en distintos modelos

¿Es posible aceptar más rápido si se usa cintas adicionales?

Complejidad en distintos modelos

¿Es posible aceptar más rápido si se usa cintas adicionales?

Teorema

Si un lenguaje L es aceptado por una MT M_1 con k cintas ($k \geq 2$) en tiempo $O(t(n))$, entonces L es aceptado por una MT M_2 con una cinta en tiempo $O(t(n)^2)$.

Complejidad en distintos modelos

¿Es posible aceptar más rápido si se usa cintas adicionales?

Teorema

Si un lenguaje L es aceptado por una MT M_1 con k cintas ($k \geq 2$) en tiempo $O(t(n))$, entonces L es aceptado por una MT M_2 con una cinta en tiempo $O(t(n)^2)$.

Ejercicio

Demuestre el teorema.

- ¿Es posible reducir la diferencia entre M_1 y M_2 ?

Complejidad en distintos modelos

Sea $L = \{w \in \{0, 1, \#\}^* \mid w \text{ es un palíndromo}\}$.

- ▶ L es aceptado por una MT con dos cintas en tiempo $O(n)$
- ▶ ¿Puede ser L aceptado en tiempo lineal por una MT con una cinta?

Complejidad en distintos modelos

Sea $L = \{w \in \{0, 1, \#\}^* \mid w \text{ es un palíndromo}\}$.

- ▶ L es aceptado por una MT con dos cintas en tiempo $O(n)$
- ▶ ¿Puede ser L aceptado en tiempo lineal por una MT con una cinta?

Proposición

Sea M una MT con una cinta. Si $L = L(M)$, entonces M funciona en tiempo $\Omega(n^2)$.

Complejidad en distintos modelos

Sea $L = \{w \in \{0, 1, \#\}^* \mid w \text{ es un palíndromo}\}$.

- ▶ L es aceptado por una MT con dos cintas en tiempo $O(n)$
- ▶ ¿Puede ser L aceptado en tiempo lineal por una MT con una cinta?

Proposición

Sea M una MT con una cinta. Si $L = L(M)$, entonces M funciona en tiempo $\Omega(n^2)$.

Demostración: Suponga que $L = L(M)$, donde M es una MT con una cinta.

- ▶ Sea Q el conjunto de estados de M

Complejidad en distintos modelos

Sin pérdida de generalidad, suponemos que M siempre recorre toda la palabra de entrada.

- ¿Por qué podemos suponer esto?

Para $w \in \{0, 1, \#\}^*$, sea w^r la palabra obtenida al escribir w en el sentido inverso.

Defina L_n como el siguiente lenguaje (para $n > 0$ divisible por 4):

$$L_n = \{w\#^{\frac{n}{2}}w^r \mid w \in \{0, 1\}^{\frac{n}{4}}\}.$$

Nótese que $L_n \subseteq L$.

Complejidad en distintos modelos

Sea $w \in L_n$ y $\frac{n}{4} \leq i \leq \frac{3n}{4}$. Entonces $C_i(w)$ es la secuencia de estados $[q_1, \dots, q_k]$ en que se encuentra M después de moverse entre las posiciones i e $i+1$ (en cualquiera de las dos direcciones) en la ejecución que tiene a w como entrada.

Sea $C(w) = \{C_i(w) \mid \frac{n}{4} \leq i \leq \frac{3n}{4}\}$

Lema

Si $w_1, w_2 \in L_n$ y $w_1 \neq w_2$, entonces $C(w_1) \cap C(w_2) = \emptyset$.

Demostración: Suponga que el lema es falso. Entonces existen $i, j \in \{\frac{n}{4}, \dots, \frac{3n}{4}\}$ tales que $C_i(w_1) = C_j(w_2)$.

Sean u_1 y u_2 las palabras formadas por los primeros i símbolos de w_1 y los últimos $n - j$ símbolos de w_2 , respectivamente.

Dado que $C_i(w_1) = C_j(w_2)$, se tiene que $u_1 u_2$ es aceptado por M .

► ¿Cómo se demuestra esto?

Pero $u_1 u_2$ no es un palíndromo, por lo que tenemos una contradicción.



Complejidad en distintos modelos

Para $w \in L_n$, sea s_w la secuencia más corta en $C(w)$.

$$\blacktriangleright S_n = \{s_w \mid w \in L_n\}$$

Por el lema sabemos que $s_{w_1} \neq s_{w_2}$ si $w_1 \neq w_2$.

$$\blacktriangleright \text{Por lo tanto: } |S_n| = |L_n| = 2^{\frac{n}{4}}$$

Sea m el largo de la secuencia mas larga en S_n .

$\blacktriangleright$ Cantidad de posibles secuencias de largo a lo más m :

$$\sum_{i=0}^m |Q|^i = \frac{|Q|^{m+1} - 1}{|Q| - 1}$$

Complejidad en distintos modelos

De lo anterior concluimos que: $\frac{|Q|^{m+1}-1}{|Q|-1} \geq 2^{\frac{n}{4}}$.

► ¿Por qué?

Se tiene entonces que m es $\Omega(n)$.

► Por lo tanto existe $w_0 \in L_n$ para el cual $|s_{w_0}|$ es $\Omega(n)$.

Entonces: Todas las secuencias en $C(w_0)$ son de largo $\Omega(n)$.

Conclusión: Con entrada w_0 , la máquina M toma tiempo $\Omega(n^2)$.

► Puesto que M tiene que generar $\frac{n}{2}$ secuencias de estados de largo $\Omega(n)$



Un ingrediente necesario: No determinismo

Definición

Máquina de Turing no determinista: $(Q, \Sigma, \Gamma, q_0, \delta, F)$

- ▶ Q es un conjunto finito de estados
- ▶ Σ es un alfabeto finito tal que $\vdash, \sqcup \notin \Sigma$
- ▶ Γ es un alfabeto finito tal que $\Sigma \cup \{\vdash, \sqcup\} \subseteq \Gamma$
- ▶ $q_0 \in Q$ es el estado inicial
- ▶ $F \subseteq Q$ es un conjunto de estados finales
- ▶ δ es una relación de transición:

$$\delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{\leftarrow, \sqcup, \rightarrow\}$$

Máquinas de Turing no determinista: Funcionamiento

La inicialización es igual que para el caso determinista.

- ▶ Al comenzar a funcionar, la máquina se encuentra en el estado q_0 y su cabeza lectora está en la posición 1 de la cinta

En cada instante la máquina se encuentra en un estado q y su cabeza lectora está en una posición p que contiene un símbolo a .

- ▶ Sea $T = \{(q', b, X) \mid (q, a, q', b, X) \in \delta\}$. Si $T \neq \emptyset$, entonces la máquina elige $(q', b, X) \in T$ y:

Máquinas de Turing no determinista: Funcionamiento

La inicialización es igual que para el caso determinista.

- ▶ Al comenzar a funcionar, la máquina se encuentra en el estado q_0 y su cabeza lectora está en la posición 1 de la cinta

En cada instante la máquina se encuentra en un estado q y su cabeza lectora está en una posición p que contiene un símbolo a .

- ▶ Sea $T = \{(q', b, X) \mid (q, a, q', b, X) \in \delta\}$. Si $T \neq \emptyset$, entonces la máquina elige $(q', b, X) \in T$ y:
 - ▶ escribe el símbolo b en la posición p de la cinta

Máquinas de Turing no determinista: Funcionamiento

La inicialización es igual que para el caso determinista.

- ▶ Al comenzar a funcionar, la máquina se encuentra en el estado q_0 y su cabeza lectora está en la posición 1 de la cinta

En cada instante la máquina se encuentra en un estado q y su cabeza lectora está en una posición p que contiene un símbolo a .

- ▶ Sea $T = \{(q', b, X) \mid (q, a, q', b, X) \in \delta\}$. Si $T \neq \emptyset$, entonces la máquina elige $(q', b, X) \in T$ y:
 - ▶ escribe el símbolo b en la posición p de la cinta
 - ▶ cambia de estado desde q a q'

Máquinas de Turing no determinista: Funcionamiento

La inicialización es igual que para el caso determinista.

- ▶ Al comenzar a funcionar, la máquina se encuentra en el estado q_0 y su cabeza lectora está en la posición 1 de la cinta

En cada instante la máquina se encuentra en un estado q y su cabeza lectora está en una posición p que contiene un símbolo a .

- ▶ Sea $T = \{(q', b, X) \mid (q, a, q', b, X) \in \delta\}$. Si $T \neq \emptyset$, entonces la máquina elige $(q', b, X) \in T$ y:
 - ▶ escribe el símbolo b en la posición p de la cinta
 - ▶ cambia de estado desde q a q'
 - ▶ mueve la cabeza lectora a la posición $p - 1$ si X es $\leftarrow$, y a la posición $p + 1$ si X es $\rightarrow$. Si X es $\square$, entonces la cabeza lectora permanece en la posición p

Máquinas de Turing no deterministas: Aceptación

Una palabra w es aceptada por una MT no determinista M si y sólo si **existe** una ejecución de M con entrada w que se detiene en un estado de F .

Definición

Lenguaje aceptado por una MT no determinista M :

$$L(M) = \{w \in \Sigma^* \mid M \text{ acepta } w\}$$

¿Es posible aceptar más lenguajes con las MTs no deterministas?

¿Es posible aceptar más lenguajes con las MTs no deterministas?

Teorema

Si un lenguaje L es aceptado por una MT no determinista M_1 , entonces L es aceptado por una MT determinista M_2 .

Determinismo vs no determinismo

¿Es posible aceptar más lenguajes con las MTs no deterministas?

Teorema

Si un lenguaje L es aceptado por una MT no determinista M_1 , entonces L es aceptado por una MT determinista M_2 .

Ejercicio

Demuestre el teorema.

- ▶ ¿Cuál es la diferencia de complejidad entre M_1 y M_2 ?

Máquinas de Turing no deterministas: Complejidad

Para una MT no determinista:

- ▶ **Paso de M :** Ejecutar una instrucción de la **relación** de transición
- ▶ **$tiempo_M(w)$:** Número de pasos de M con entrada w en la ejecución más corta que acepta a w

Sólo está definido para palabras aceptadas por M

- ▶ ¿Por qué?

Definición

El tiempo de funcionamiento de una MT no determinista M en el peor caso es definido por la función t_M :

$$t_M(n) = \text{máx} \left[\{n\} \cup \{ \text{tiempo}_M(w) \mid w \in \Sigma^*, \right. \\ \left. |w| = n \text{ y } M \text{ acepta a } w \} \right]$$

Definición

El tiempo de funcionamiento de una MT no determinista M en el peor caso es definido por la función t_M :

$$t_M(n) = \max \left[\{n\} \cup \{ \text{tiempo}_M(w) \mid w \in \Sigma^*, \right. \\ \left. |w| = n \text{ y } M \text{ acepta a } w \} \right]$$

¿Por qué incluimos $\{n\}$ en la definición?

Espacio utilizado en una Máquina de Turing

Para introducir la noción de espacio utilizado en una MT tenemos que distinguir entre:

- ▶ el espacio ocupado por la entrada, y
- ▶ el espacio necesario para procesar la entrada

Para hacer esta distinción utilizamos MT con dos cintas:

- ▶ **Cinta para la entrada:** Sólo de lectura
- ▶ **Cinta de trabajo:** Como en una MT usual

El espacio utilizado se mide en términos de las celdas visitadas en la cinta de trabajo.

Definición

Máquina de Turing (determinista) con cinta de trabajo:
 $(Q, \Sigma, \Gamma, q_0, \delta, F)$

- ▶ Q es un conjunto finito de estados.
- ▶ Σ es un alfabeto tal que $\vdash, B \notin \Sigma$.
- ▶ Γ es un alfabeto tal que $\Sigma \cup \{\vdash, B\} \subseteq \Gamma$.
- ▶ $q_0 \in Q$ es el estado inicial.
- ▶ $F \subseteq Q$ es un conjunto de estados finales.
- ▶ δ es una función parcial:

$$\delta : Q \times \Gamma \times \Gamma \rightarrow Q \times \{\leftarrow, \square, \rightarrow\} \times \Gamma \times \{\leftarrow, \square, \rightarrow\}$$

Espacio utilizado en una MT

Dada una MT determinista M con cinta de trabajo que para en todas las entradas:

- ▶ $\text{espacio}_M(w)$: Número de celdas distintas visitadas en la cinta de trabajo de M al procesar w

Espacio utilizado en una MT

Dada una MT determinista M con cinta de trabajo que para en todas las entradas:

- ▶ $\text{espacio}_M(w)$: Número de celdas distintas visitadas en la cinta de trabajo de M al procesar w

Definición

Dado un número natural n :

$$s_M(n) = \text{máx}\{ \text{espacio}_M(w) \mid w \in \Sigma^* \text{ y } |w| = n \}$$

Espacio utilizado en una MT

Dada una MT determinista M con cinta de trabajo que para en todas las entradas:

- ▶ $\text{espacio}_M(w)$: Número de celdas distintas visitadas en la cinta de trabajo de M al procesar w

Definición

Dado un número natural n :

$$s_M(n) = \text{máx}\{ \text{espacio}_M(w) \mid w \in \Sigma^* \text{ y } |w| = n \}$$

s_M : Espacio utilizado en la ejecución de M en el **peor caso**

Espacio utilizado en una MT

Un lenguaje L puede es aceptado por una MT M con cinta de trabajo en **espacio** $O(s(n))$ si $L = L(M)$ y $s_M(n)$ es $O(s(n))$.

Espacio utilizado en una MT

Un lenguaje L puede es aceptado por una MT M con cinta de trabajo en **espacio** $O(s(n))$ si $L = L(M)$ y $s_M(n)$ es $O(s(n))$.

Ejemplo

¿Cuál es el espacio requerido para aceptar el lenguaje $L = \{w \in \{0,1\}^* \mid w \text{ es un palíndromo}\}$?

- ¿Cuadrático? ¿Lineal? ¿Logarítmico? ¿Constante?

Ejercicio

Defina la noción de MT no determinista con una cinta de trabajo.

Dada una MT no determinista con cinta de trabajo M :

- ▶ $\text{espacio}_M(w)$: Número mínimo de celdas distintas visitadas en la cinta de trabajo de M , entre todas las ejecuciones de M que aceptan w

Sólo está definida para palabras aceptadas por M

Espacio utilizado en una MT: No determinismo

Definición

Dado un número natural n :

$$s_M(n) = \max \left[\{1\} \cup \{ \text{espacio}_M(w) \mid w \in \Sigma^*, \right. \\ \left. |w| = n \text{ y } M \text{ acepta } w \} \right]$$

¿Por que incluimos $\{1\}$ en la definición?