Práctico 1 Complejidad Computacional

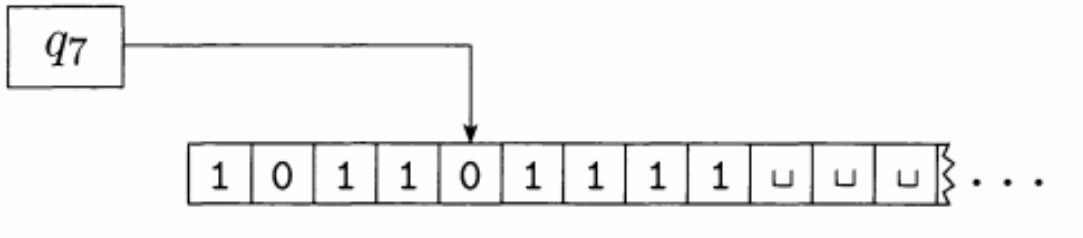
Julián Viera

jviera@fing.edu.uy
julviera44@gmail.com

10 de agosto de 2021

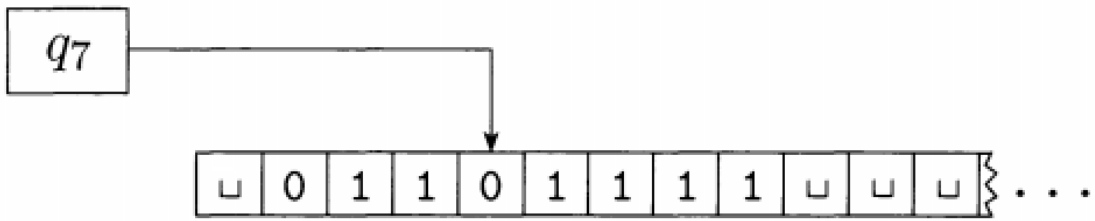
Agenda

Descripción informal del modelo de la MT de Sipser



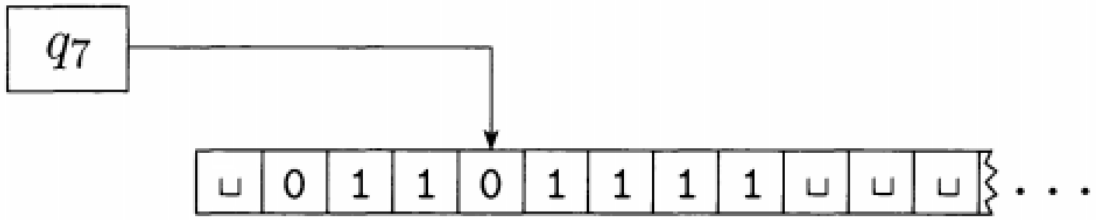
- Modelo de una cinta semi-infinita unidimensional.
- La cinta está compuesta de celdas, en donde se pueden leer y escribir símbolos. La cinta se extiende infinitamente a la derecha a partir de una celda inicial.
- Inicialmente la cinta contiene solo la tira de entrada en las celdas iniciales, y el resto de la celdas contienen un símbolo especial (blanco, "blank" o $\not b$),denotado en la figura como $\sqcup$.
- Hay un cabezal móvil (header) que señala la celda actual de la cinta.
- El cabezal puede moverse una posición ya sea a la izquierda como a la derecha (se mueve a una celda vecina). Si está en la celda inicial no se mueve a la izquierda, en este caso solo puede moverse a la derecha.

Descripción informal del modelo de la MT de Sipser



- Para indicar que el cabezal está apuntando a la celda inicial, generalmente se marca dicha celda con un símbolo especial (por ejemplo un blanco). Esto permite reconocer el comienzo de la cinta.
- La MT tiene un número finito de estados, y en cada momento se encuentra en un único estado.
- La MT tiene dos estados especiales: de aceptación ("accept") y de rechazo ("reject"). Si la MT llega a uno de estos dos estados, finaliza su ejecución (son estados de parada de la MT).

Funcionamiento de la MT



En cada paso de su operación, la MT ejecuta las siguientes acciones:

1. Lee el símbolo escrito en la celda apuntada por el cabezal (header)
2. En base a su estado actual y al símbolo leído, computa el nuevo estado, el nuevo símbolo a escribir en la celda actual, y la dirección de su próximo movimiento
3. Escribe el nuevo símbolo en la celda actual (eventualmente puede ser el mismo y dejarla incambiada)
4. Mueve el cabezal en la dirección computada (se desplaza a la celda de la izquierda o hacia la celda de la derecha)

Descripción formal del modelo de la MT de Sipser

La MT es una tupla de siete elementos $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ donde:

- Q conjunto finito de estados
- Σ alfabeto finito de entrada (no contiene el símbolo blanco $\not b$)
- Γ alfabeto finito de la cinta ($\not b \in \Gamma, \Sigma \subset \Gamma$)
- δ función de transición de la MT: $Qx\Gamma \rightarrow Qx\Gamma x(L, R)$
- $q_0 \in Q$ estado inicial
- $q_{accept} \in Q$ estado de aceptación
- $q_{reject} \in Q$ estado de rechazo ($q_{reject} \neq q_{accept}$)

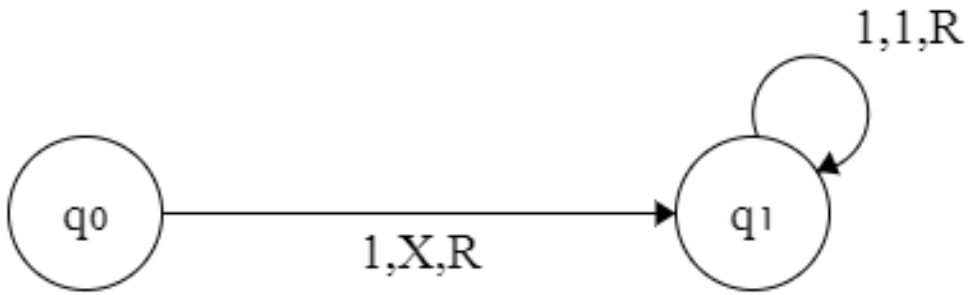
Descripción a nivel de implementación de una MT

Consiste en describir, en lenguaje natural, la serie de pasos que ejecuta la MT, partiendo desde el estado inicial q_0 hasta llegar a los estados de aceptación o rechazo.
Es una descripción de alto nivel, del tipo pseudocódigo algoritmico, de las acciones de la MT. En general se numeran los pasos, y pueden existir transferencias de ejecución de una paso hacia otro.
Algunos ejemplos de sentencias de la descripción de implementación:

- "Recorrer la cinta hacia la derecha hasta encontrar el símbolo S"
- "Sustituir el símbolo A por el símbolo B"
- "Volver al extremo izquierdo de la cinta"
- "Leer el simbolo en la celda actual. Si es A, ir al paso 3), si es B, ir al paso 7)".

Diagrama de estados de la MT

Cuando se reguiere dar la descripción formal de una MT, la función de transición δ se especifica a través de un diagrama de estados. En dicho diagrama aparecen todos los estados de la MT, y las transiciones entre los mismos se representan por flechas etiquetadas. Cada etiqueta muestra el símbolo de la celda actual apuntada por el cabezal, el nuevo símbolo a escribir en dicha celda y la dirección del movimiento siguiente del cabezal (L o R).
A modo de ejemplo, en la figura que sigue estando la MT en el estado q_0 y con la celda actual conteniendo un 1, se "marca" dicha celda escribiendo una X y se mueve el cabezal a la derecha, entrando en el estado q_1.



Ejercicio 2 - Planteo del problema

Dar una descripción a nivel de implementación y una descripción formal del siguiente lenguaje:
$B = \{x \in \{0,1,2\}^* \ / \ x = 0^n 1^n 2^n, \text{ con } n \geq 0\}$

Ejemplos de tiras de B :

- $\epsilon \quad (n = 0)$
- $012 \quad (n = 1)$
- $001122 \quad (n = 2)$

Ejercicio 2 - idea para la solución

$$B = \{x \in \{0,1,2\}^* \ / \ x = 0^n 1^n 2^n, \text{ con } n \geq 0\}$$

- Por cada 0 en la tira de entrada, chequear que haya tanto un 1 como un 2 que le correspondan en la tira.
- Se van marcando las ternas (0,1,2) encontradas.
- Si para un 0 dado falta el 1 o el 2, la MT debe rechazar.
- Si se terminan de marcar todos los ceros de la entrada y no hay símbolos sin marcar $\Rightarrow$ la tira pertence al lenguaje, y la MT debe aceptar.
- Si se terminan de marcar todos los ceros de la entrada pero todavía hay 1's ó 2's en la cinta sin marcar, $\Rightarrow$ la tira no pertence al lenguaje, y la MT debe rechazar.

Ejercicio 2 - descripción a nivel implementación

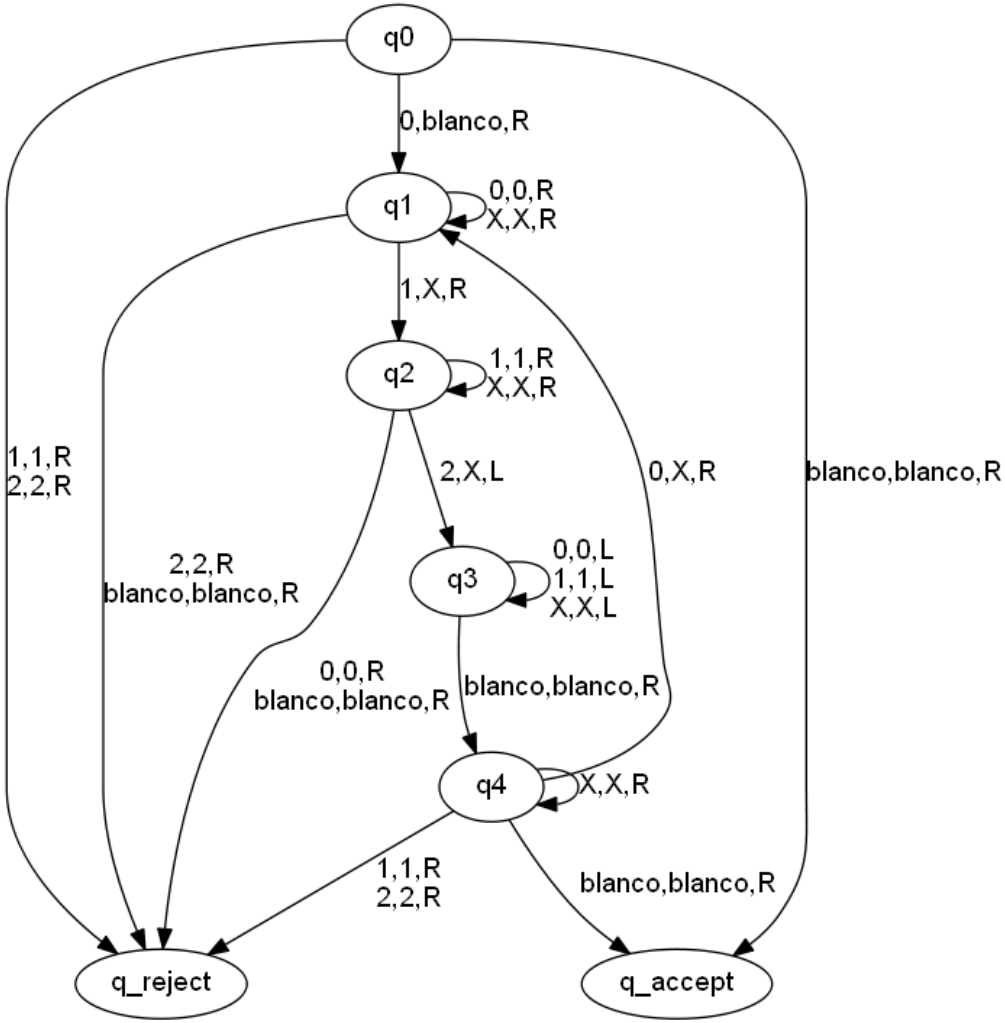
$B = \{x \in \{0,1,2\}^* \ / \ x = 0^n 1^n 2^n, \text{ con } n \geq 0\}$

Lo que sigue es una refinación de la idea general descrita anteriormente.

1. Analizar el primer símbolo en la cinta. Si es ␣,aceptar. Si es 1 ó 2, rechazar. Si es 0, sustituir por ␣ (marca de inicio de cinta).
2. Recorrer la cinta hacia la derecha buscando un 1. Ignorar 0's y X's. Si se encuentra fin de entrada (␣) ó 2, rechazar. Si se encuetra 1, sustituir por X.
3. Recorrer la cinta hacia la derecha buscando un 2. Ignorar 1's y X's. Si se encuentra 0 ó ␣, rechazar. Si se encuetra 2, sustituir por X.
4. Volver al extremo izquierdo de la cinta.
5. Recorrer la cinta hacia la derecha, ignorando X's. Si se encuentra 1 ó 2, rechazar. Si se encuentra ␣, aceptar. Si se encuentra 0,sustituir por X e ir al paso 2.

Ejercicio 2 - diagrama de estados de la MT

$$B = \{x \in \{0,1,2\}^* \ / \ x = 0^n 1^n 2^n, \text{ con } n \geq 0\}$$



Ejercicio 4 - Planteo del problema

MT con LEFT RESET M_{LR}
En M_{LR}, $\delta : Qx\Gamma \rightarrow Qx\Gamma x\{R, RESET\}$
El movimiento *RESET* lleva el cabezal al extremo izquierdo de la cinta. Esta MT no tiene la posibilidad de mover el cabezal a la celda que está a la izquierda de la celda actual.

Sea M una MT convencional. El problema consiste en probar que M puede simular a M_{LR} y recíprocamente, es decir que los lenguajes reconocidos por ambas máquinas son los mismos.

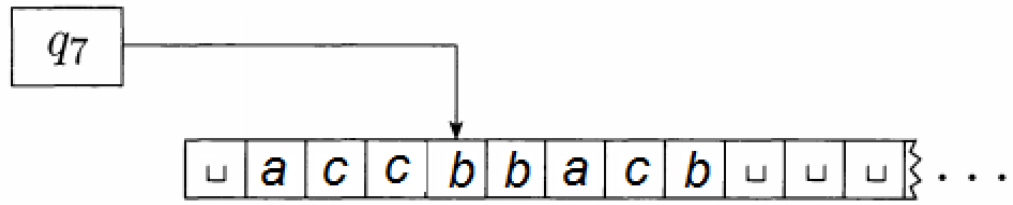
Ejercicio 4 - una posible solución

$\Rightarrow$ M simula a M_{LR}:
Se verifica en forma trivial. Cuando M_{LR} hace $RESET$, M se mueve repetidamente a la izquierda con L hasta detectar el extremo izquierdo de la cinta.

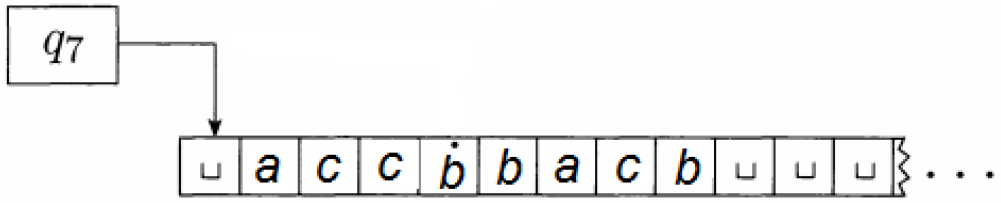
$\Leftarrow$ M_{LR} simula a M:
Hay que probar que dado M, puedo encontar una M_{LR} que puede simular el movimiento a la izquierda (L) de M.
Como tenemos libertad para elegir el alfabeto de M_{LR}, lo elegimos de forma que contenga al alfabeto de M y además por cada símbolo σ del alfabeto de M, en el de M_{LR} se tiene el símbolo $\dot{\sigma}$.

Ejercicio 4 - una posible solución

Vamos a presentar un algoritmo para hacer que M_{LR} pueda moverse una celda hacia la izquierda, ejemplificando el paso a paso con la configuración inicial de la MT que se muestra en la figura:



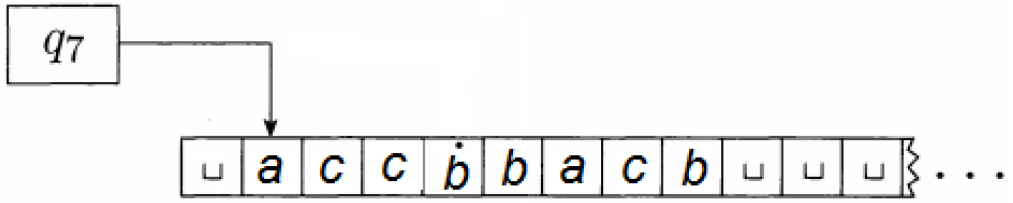
1. Marcar con punto el símbolo apuntado por el cabezal de M_{LR}. Hacer *RESET*.



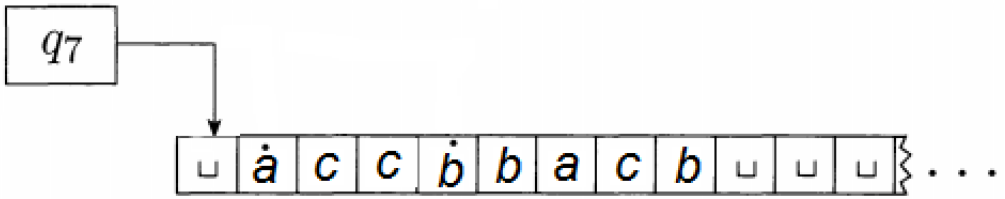
2. Si la celda del extremo izquierdo tiene punto (es el blanco con punto arriba), borrarlo y *TERMINAR*. Ya se estaba inicialmente en el extremo izquierdo de la cinta y la MT no se puede mover a la izquierda en esta posición.

Ejercicio 4 - una posible solución

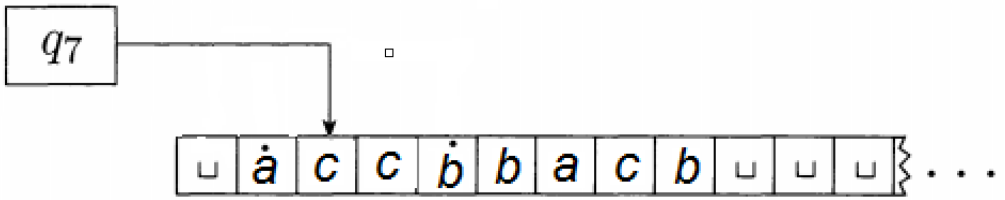
3. Mover el cabezal un lugar hacia la derecha con R.



4. Si la celda tiene símbolo con punto, la casilla destino es el extremo izquierdo de la cinta. Borrar el punto. Hacer *RESET* y *TERMINAR*.
5. Marcar con punto. Hacer *RESET*.

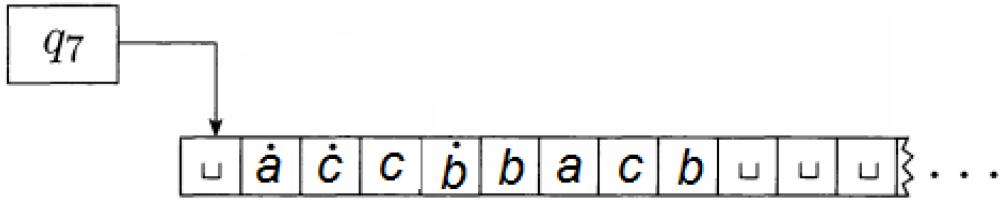


6. Moverse dos celdas a la derecha con R.

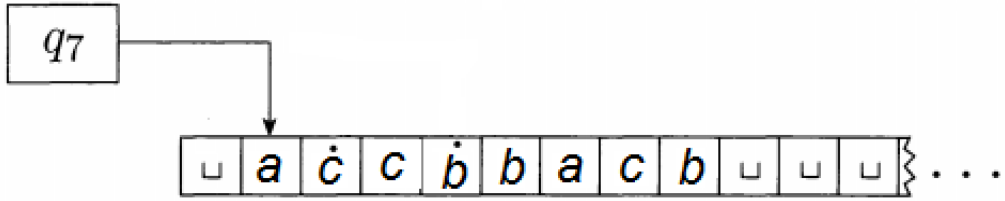


Ejercicio 4 - una posible solución

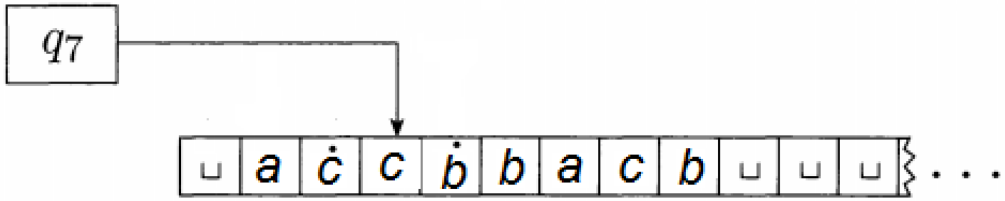
7. Si la celda actual no tiene punto, marcarla con punto y hacer *RESET*. Si tiene punto, ir al paso 9).



8. Mover el cabezal hasta la primer celda marcada con punto y borrarlo. Ir al paso 6).

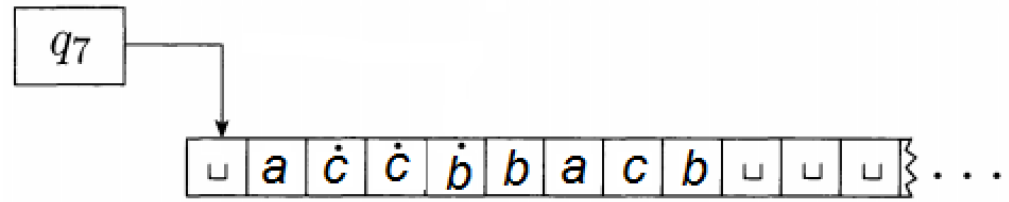


6. Moverse dos celdas a la derecha con R.

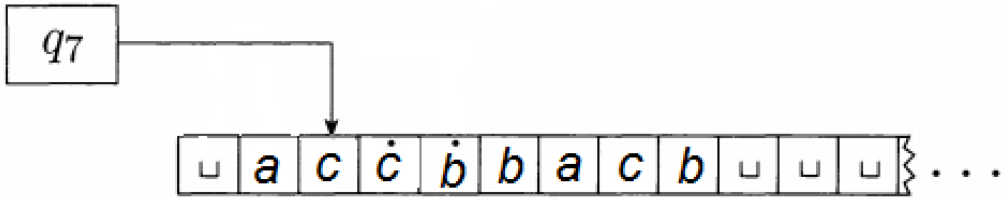


Ejercicio 4 - una posible solución

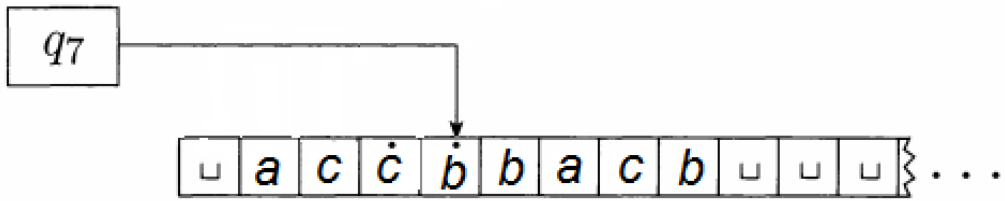
7. Si la celda actual no tiene punto, marcarla con punto y hacer *RESET*. Si tiene punto, ir al paso 9).



8. Mover el cabezal hasta la primer celda marcada con punto y borrarlo. Ir al paso 6).



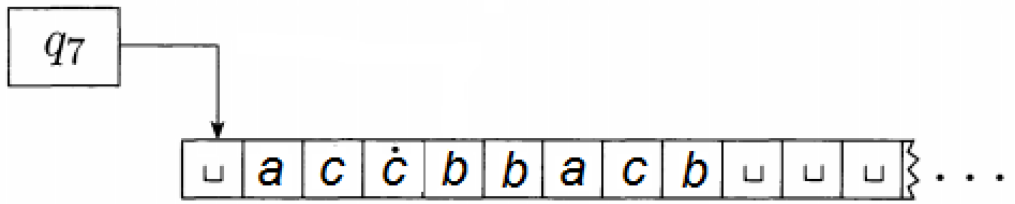
6. Moverse dos celdas a la derecha con R.



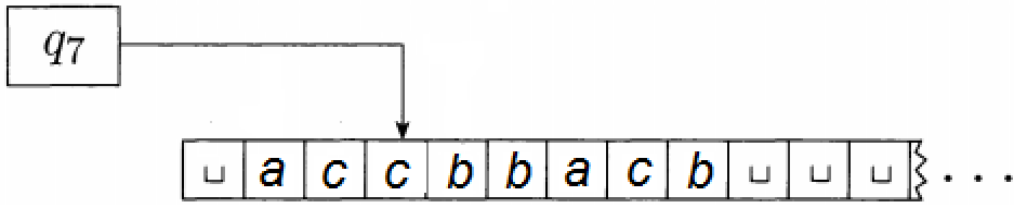
Ejercicio 4 - una posible solución

7. Si la celda actual no tiene punto, marcarla con punto y hacer *RESET*. Si tiene punto, ir al paso 9).

9. Se llegó a la posición original del cabezal. Sacarle el punto al símbolo. Hacer *RESET*.



10. Moverse con R hasta la primera celda con punto, que es la celda destino. Sacarle el punto y *TERMINAR*.



Ejercicio 4 - Overhead en tiempo de M_{LR}

Partiendo de la hipótesis de que la MT convencional M tiene un tiempo de ejecución $T(n)$ para una entrada de tamaño n, interesa saber cual es el sobrecosto (overhead) en el tiempo que tiene hacer ese mismo cómputo con la máquina M_{LR}, es decir estimar cuantos pasos adicionales de ejecución precisa realizar M_{LR} respecto a M.
Para eso necesitamos contar primero el número de pasos que hace M_{LR} para moverse un lugar a la izquierda, es decir para simular el movimiento L de M, asumiendo que el cabezal está inicialmente en la celda n.
Analizando el algoritmo, se comprueba que esta cantidad de pasos, que llamaremos T_L,se compone de:

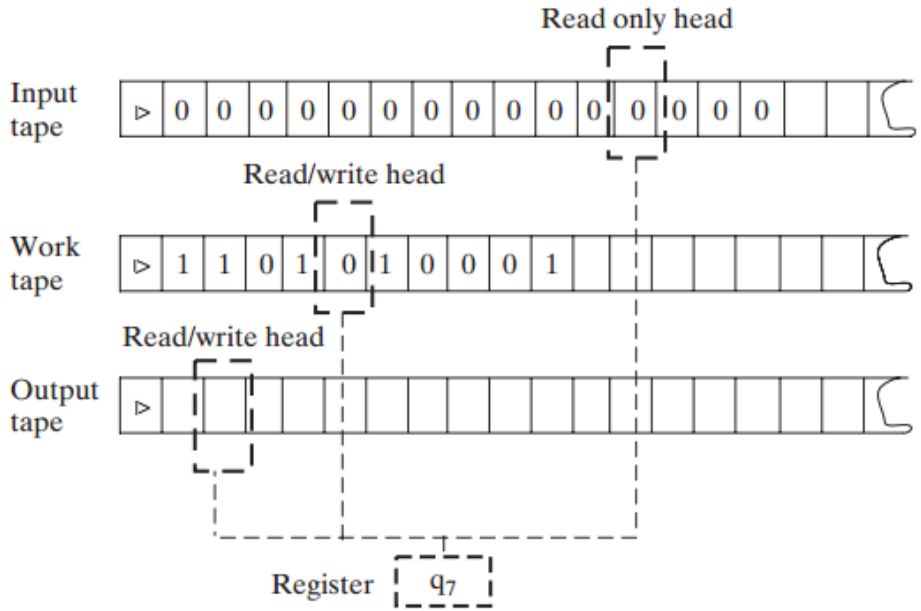
- Un paso (movimiento) *RESET* inicial.
- Por cada celda intermedia en el lugar i entre el comienzo de la cinta y la posición inicial n, el algoritmo hace i movimientos R hacia la derecha y un *RESET* $\Rightarrow i+1$ pasos.
- Al final hace $n-1$ movimientos R para llegar a la celda $n-1$.

Ejercicio 4 - Overhead en tiempo de M_{LR}

$$\Rightarrow T_L(n) = 1 + \sum_{i=1}^{n}(i+1) + n - 1 = \sum_{i=1}^{n} i + 2n = \frac{n^2}{2} + \frac{5n}{2}$$

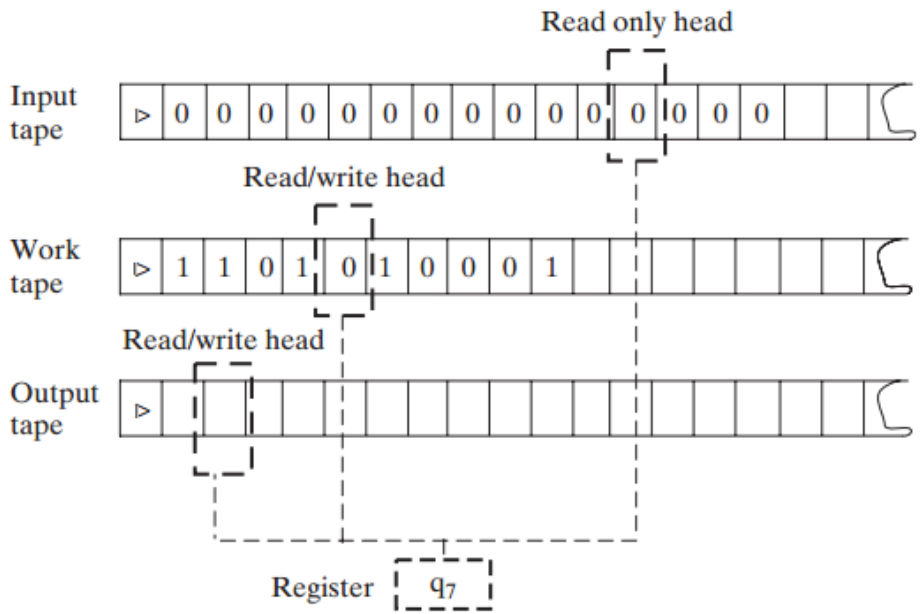
La observación clave es que, como la MT M por hipótesis computa en $T(n)$ pasos $\Longrightarrow$ la cantidad de celdas de la cinta visitadas durante la ejecución de M es a lo sumo $T(n) \Longrightarrow$ la celda más a la derecha que se puede alcanzar en M está en la posición $T(n)$.
Esto implica que el tiempo máximo T_L que necesita M_{LR} para emular un movimiento a la izquierda vale
$T_L(T(n)) = \frac{T(n)^2}{2} + \frac{5T(n)}{2} = \mathcal{O}(T(n)^2)$
Un peor caso de tiempo de ejecución de M_{LR} consiste en que todos los pasos de M sean moverse a la izquierda partiendo de la posición $T(n)$. Como a lo sumo M ejecuta $T(n)$ pasos $\Rightarrow$ el tiempo de ejecución de M_{LR} es $\mathcal{O}(T(n)^3)$
$\Rightarrow$ El *OVERHEAD* del tiempo de ejecución de M_{LR} respecto a M es **polinomial**.

Descripción informal del modelo de la MT de Arora-Barak



- Modelo de k cintas semi-infinitas unidimensionales, con celda inicial conteniendo el símbolo especial *start* (▷).
- Hay una cinta de entrada ("input tape"), en donde sólo se puede leer la tira de entrada a la MT, es del tipo read only.
- Existen $k-1$ cintas de trabajo ("working tapes")que son de tipo lectura-escritura.
- La última cinta de trabajo es la salida de la MT ("output tape"), y es donde se escribe el resultado final antes de parar.

Descripción informal del modelo de la MT de Arora-Barak

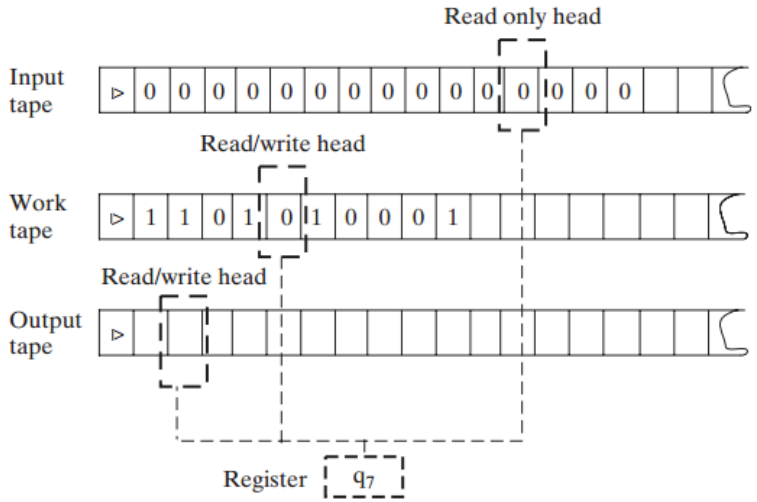


- Hay un cabezal móvil (header)para cada cinta que señala la celda actual de esa cinta.
- El cabezal de cada cinta puede moverse una única posición ya sea a la izquierda como a la derecha, o puede quedarse en la celda donde está. Si apunta a la celda inicial, no se mueve a la izquierda.
- La MT tiene un número finito de estados, y en cada momento se encuentra en un único estado, que se guarda en un registro.

Descripción informal del modelo de la MT de Arora-Barak

- Hay dos estados especiales: de inicio ("start") y de parada ("halt"). Si la MT llega al estado "halt", finaliza su ejecución.
- En el estado "start", los cabezales de todas las cintas están posicionados en el extremo izquierdo apuntando al símbolo *start* ($\triangleright$).
- En el estado "start", la cinta de entrada contiene el símbolo $\triangleright$, una tira finita de símbolos (la "entrada" de la MT), distintos del símbolo especial blanco (denotado como $\square$) y en el resto de las celdas el símbolo blanco.
- En el estado "start", todas las celdas distintas de la inicial de las $k-1$ cintas de trabajo se inicializan con un blanco.

Funcionamiento de la MT



En cada paso de su operación, la MT ejecuta lo siguiente:

1. Lee los k símbolos de la celdas apuntadas por los cabezales.
2. En base a su estado actual y a los símbolos leídos, computa el nuevo estado, los $k-1$ símbolos a escribir en las $k-1$ cintas de trabajo y la dirección del movimiento de cada una de las k cintas.
3. Escribe los nuevos símbolos en la celdas actuales de las $k-1$ cintas de trabajo.
4. Mueve los k cabezales a las posiciones computadas.

Descripción formal del modelo de la MT de Arora-Barak

La MT es una tupla de tres elementos (Q, Γ, δ) donde:

- Q conjunto finito de estados que contiene a los estados especiales q_{start} y q_{halt}.
- $q_{start} \in Q$ estado de inicio de la ejecución.
- $q_{halt} \in Q$ estado de parada. Cuando la MT llega a q_{halt} ya no se modifica más la cinta ni hay cambios de estado.
- Γ alfabeto finito de la cinta, que contiene a los simbolos especiales blanco ($\square$) y "start" ($\triangleright$) y a los números 0 y 1.
- δ función de transición de la MT.
  $\delta : Q x \Gamma^k \rightarrow Q x \Gamma^{k-1} x (L, S, R)^k$, con $k \geq 2$.

Función computada por una MT M

Sea $f : \{0,1\}^* \to \{0,1\}^*$

Decimos que la MT M **computa** f si siempre que M se inicializa con una tira de entrada $x \in \{0,1\}^*$ en su cinta de entrada, M *para* con $f(x)$ escrita en su cinta de salida.

Tiempo de cómputo de una MT M

Sean Sea $f : \{0,1\}^* \to \{0,1\}^*$ y $T : N \to N$ una función dada. Se dice que la la MT M computa f en tiempo $T(n)$, si la computación para cualquier tira de entrada x con $|x| = n$ requiere **a lo sumo** $T(n)$ pasos de ejecución de M.

Es importante resaltar en esta definición, que se está asegurando que la MT M termina su computación en un número de pasos $T(n)$ **para toda** entrada x de largo n, es decir para cada una de las 2^n entradas posibles de largo n.

Función time-constructible M

Una función $T : N \to N$ es **time-constructible** si $T(n) \geq n$ y existe una MT M tal que dada una entrada x con $|x| = n$ (por ejemplo 1^n), M computa la función $x \to \llcorner T(n) \lrcorner$ (representación **binaria** de $T(n)$) en tiempo $\mathcal{O}(T(n))$.

Ejercicio 6 - Planteo del problema

Probar que la función $T(n) = n$ es time-constructible.
Asumir para la prueba que $n = 2^p \quad p \geq 0$.

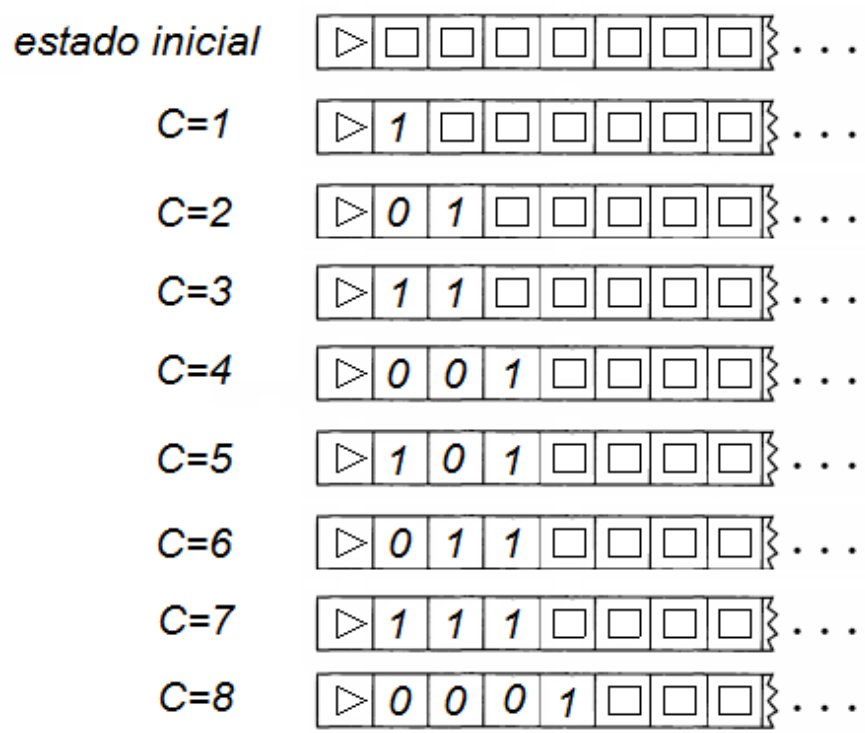
El problema consiste en mostrar que existe una MT M que con 1^n en su cinta de entrada, computa y escribe en su cinta de salida la representación binaria de n (denotada como $\llcorner n \lrcorner$)en tiempo $\mathcal{O}(n)$.

Ejercicio 6 - una posible solución

Utilizamos un contador C que se implementa en una de las cintas de trabajo de la MT M.
Este contador va a contener al final la representación binaria de n, pero con el LSB a la izquierda.
Observamos que el número de bits de $\llcorner n \lrcorner$ que se precisa almacenar en C es $p = \log n + 1$.
El algoritmo de cálculo de dicha representación binaria es el siguiente:

1. Recorrer la cinta de entrada con los n unos desde el principio hasta el final.
2. Por cada 1 de la entrada, incrementamos en 1 el contador C.
3. Al terminar, escribir el contenido de C invertido en la cinta de salida de M.

Ejercicio 6 - ejemplo de generación de $\llcorner n \lrcorner$ para $n = 8$



- En cada incremento de C el cabezal de la cinta se mueve a la derecha desde el extremo izquierdo $\triangleright$ hasta la última celda que cambia, y luego se mueve hacia la izquierda hasta la celda $\triangleright$.
- Lo anterior implica que por cada incremento de C la MT M efectúa múltiples pasos individuales.

Ejercicio 6 - conteo de cantidad de pasos

- La cantidad de movimientos hacia la derecha (MD)que ejecuta el cabezal de la cinta del contador C coincide con el número total de celdas que "cambian de estado" cada vez que se incrementa en 1 el contador C.
- Considerando la primer columna de celdas del ejemplo, correspondiente a la celda 1 de C, se observa que dicha celda cambia de estado (se recorre) en cada incremento de C, o sea n veces.
- La celda 2 de C cambia de estado $\frac{n}{2}$ veces.
- En general la celda i-ésima se visita $\frac{n}{2^{i-1}}$ veces, $i = 1..\log n + 1$.

$$\Rightarrow MD = \sum_{i=1}^{\log n+1} \frac{n}{2^{i-1}} = \sum_{j=0}^{\log n} \frac{n}{2^j} \leq n \sum_{j=0}^{\infty} \left(\frac{1}{2}\right)^j \leq 2n$$

Ejercicio 6 - conteo de cantidad de pasos

- La cantidad de movimientos hacia la izquierda (MI)que ejecuta el cabezal de la cinta del contador C coincide con la cantidad de movimientos hacia la derecha MD, pues en cada incremento de C por cada celda visitada en la recorrida hacia la derecha se debe pasar de nuevo cuando el cabezal vuelve al extremo izquierdo.
  $\Rightarrow MI = MD \leq 2n$
- Adicionalmente, de necesitan $p = \log n$ pasos de ejecución para invertir C y que el MSB quede a la izquierda en la cinta de salida.

El número total de pasos ejecutados por M vale
$NP = MI + MD + \log n \leq 4n + \log n = \mathcal{O}(n)$.
$\xrightarrow{\text{definición}}$ $T(n) = n$ es time-constructible.

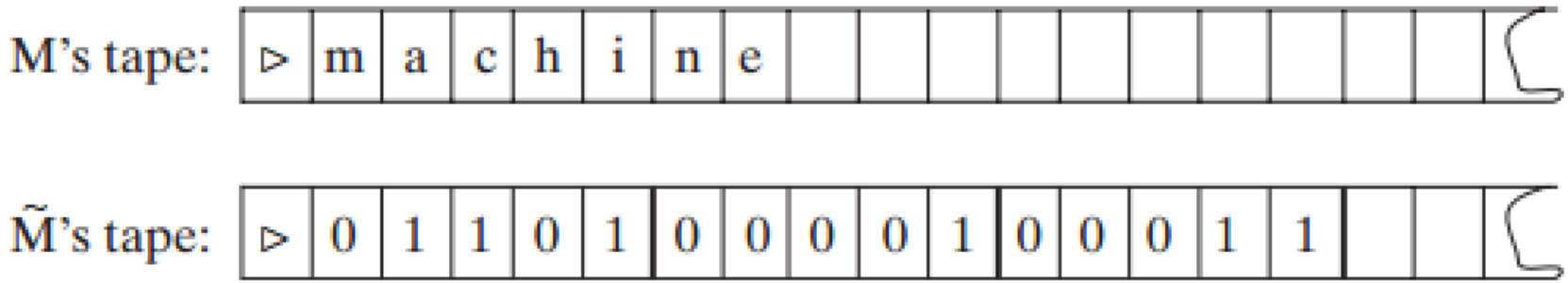
La clase de lenguajes **DTIME** (Deterministic Time)

Sea $T : N \to N$ una función.

Un lenguaje L pertenece a DTIME($T(n)$) si y solo si existe una MT M que se ejecuta en tiempo $cT(n)$, $c > 0$ y decide L.

Representación en la cinta de la MT de un simbolo σ

Sea σ un simbolo de un alfabeto Γ. Podemos representar σ con $\log |\Gamma|$ bits en una cinta de una MT convencional. En el ejemplo de la figura, se necesitan 5 bits para codificar cualquier letra del alfabeto a...z con 0's y 1's (la "m" se codifica como 01101, por ser la letra 13)



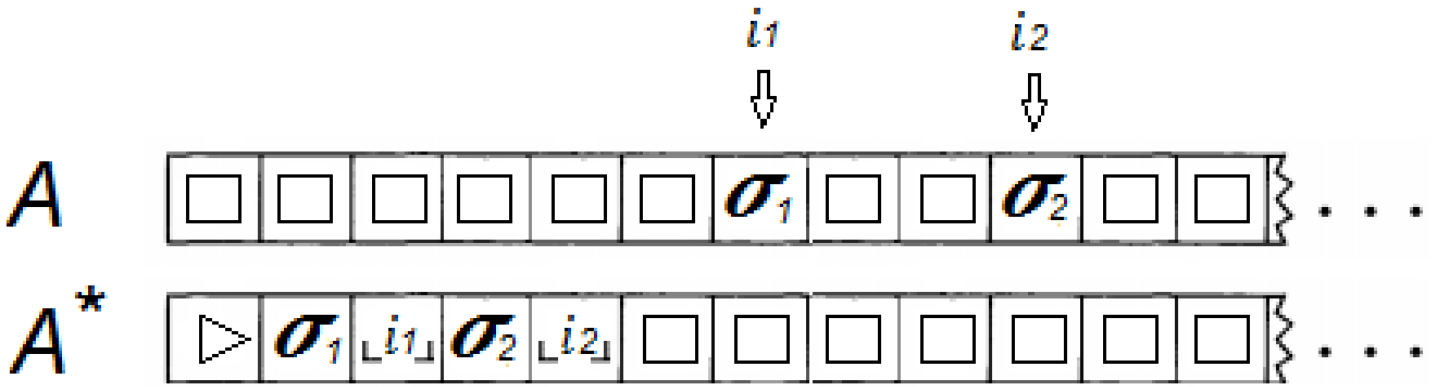
De igual modo, para codificar pares (σ, i) donde i es un entero positivo que representa el índice de un array A, la codificación de σ con $\log |\Gamma|$ bits aparece junto a $\llcorner i \lrcorner$, representación binaria de i que requiere $\log |A|$ bits, siendo $|A|$ el tamaño de A.

Ejercicio 9 - Planteo del problema (1.9 del Arora)

Se considera la variante de la MT denominada *RAM TM* (Random Access Memory Turing Machine).
Tiene un array infinito A inicializado en blancos. Hay una cinta especial de direcciones, el "adress tape".
Hay dos símbolos especiales, uno de lectura R y otro de escritura W y un estado especial q_{access}.
Si la *RAM TM* está en el estado q_{access} y en la cinta de direcciones figura $\llcorner i \lrcorner R$, entonces lee el valor de $A[i]$ y lo escribe en la celda contigua a la del símbolo R en el adress tape. Si el adress tape contiene $\llcorner i \lrcorner \sigma W$ (σ es un símbolo del alfabeto de la máquina), la máquina escribe σ en $A[i]$.
El problema pide mostrar que si una función booleana f es computable en tiempo $T(n)$ por una *RAM TM*, entonces dicha función pertenece a $DTIME(T(n)^2)$.
Esto implica mostrar que existe una MT convencional que computa f en tiempo $\mathcal{O}(T(n)^2)$.

Ejercicio 9 - una posible solución

Sea M la *RAM TM* y M^* la MT convencional de k cintas que va a simular a M.
La idea principal es que el array A de M va a estar representado en M^* por una cinta que llamaremos A^*.
Cada elemento de A^* es un par (σ, i), siendo σ un símbolo del alfabeto de M e i la posición de memoria del array A donde se almacena σ. Es decir que el array A se mapea en la cinta A^* en celdas consecutivas de A^*.



Ejercicio 9 - una posible solución

La MT M^* va a tener al igual que la MT M una cinta de direcciones D^*, y un estado q_{access}.
Simulación en M^* de la operación de lectura $\llcorner i \lrcorner R$ en M:

- M^* recorre la cinta A^* buscando el indice i.
- Si encuentra la pareja (σ, i), copia σ en la cinta de direcciones D^*.
- Si no encuentra la dirección i en A^*, escribe el símbolo blanco ($\square$) en D^*.

Simulación en M^* de la operación de escritura $\llcorner i \lrcorner W\sigma$ en M:

- M^* recorre la cinta A^* buscando el indice i.
- Si encuentra la pareja (α, i), sobreescribe α con σ.
- Si no encuentra el índice i en A^*, escribe (σ, i) al final de A^*.

Observamos que en el peor caso tanto de lectura como de escritura M^* necesita recorrer **todas** las parejas escritas en la cinta A^*.

Ejercicio 9 - Tiempo de ejecución de M^*

La observación **clave** es que como f es computada en tiempo $T(n)$ por la *RAM TM* M, en el array A de dicha máquina nunca van a haber más de $T(n)$ símbolos. Esto implica que en la cinta A^* de M^* nunca van a haber más de $T(n)$ parejas (σ, i).
Asumiendo que las operaciones dominantes en el tiempo $T(n)$ de M son las lecturas-escrituras en el array A, en cada paso de M voy a tener a lo sumo $T(n)$ pasos de M^*, considerando que haya que recorrer toda la cinta A^* en cada operación.
Como f necesita $T(n)$ pasos de ejecución en M, $\Longrightarrow$ M^* ejecuta en el peor caso $\mathcal{O}(T(n)^2)$ pasos para simular M y computar f.
$\Longrightarrow f \in \text{DTIME}\ (T(n)^2)$.