

Trabajo Práctico Nro 1 - La Máquina de Turing

Ejercicio 1

Responder breve y claramente los siguientes incisos:

1. ¿En qué se diferencia un problema de búsqueda de un problema de decisión?
 - La diferencia es que un **problema de búsqueda** consiste en encontrar una solución a una instancia dada del problema a resolver, la máquina puede devolver una solución al problema o en el caso que no exista una solución devolver "no". Mientras que un **problema de decisión** consiste en determinar la existencia de una solución a un problema, siendo las salidas posibles "si" o "no".
2. ¿Por qué en el caso de los problemas de decisión, podemos referirnos indistintamente a problemas y lenguajes?
 - Podemos referirnos indistintamente a problemas y lenguajes porque existe una **correspondencia directa y natural entre un problema de decisión y el lenguaje que representa sus instancias positivas**
 - Un problema de decisión es aquel cuya respuesta para cualquier instancia es "si" o "no".
 - Un lenguaje, en la teoría de la computación, es un conjunto de cadenas formadas a partir de un alfabeto.
 - Para un problema de decisión dado, podemos definir un lenguaje que contenga todas las cadenas que representan las instancias para las cuales la respuesta al problema es "sí". Estas son las instancias positivas del problema.
 - Una Máquina de Turing (MT) que resuelve un problema de decisión se comporta como un reconocedor de lenguajes. Cuando se le proporciona una cadena de entrada (que representa una instancia del problema):
 - Si la respuesta al problema para esa instancia es "sí", la MT acepta la cadena.
 - Si la respuesta es "no", la MT rechaza la cadena.
 - Por lo tanto, el lenguaje aceptado por la MT es precisamente el conjunto de todas las cadenas que corresponden a las instancias positivas del problema de decisión.
3. El problema de satisfactibilidad de las fórmulas booleanas, en su forma de decisión, es:
"Dada una fórmula ϕ , ¿existe una asignación A de valores de verdad que la hace verdadera?" Enunciar el problema de búsqueda asociado.

- **Problema de Búsqueda:** "Dada una fórmula booleana φ , encontrar, si existe, una asignación A de valores de verdad a las variables de φ que haga que la evaluación de φ con A sea verdadera."
4. Otra visión de MT es la que genera un lenguaje (visión generadora). En el caso del problema del inciso anterior, ¿Qué lenguaje generaría la MT de visión generadora que resuelve el problema?
- Para el problema de las fórmulas booleanas satisfactibles, el lenguaje que generaría una MT con visión generadora sería **el conjunto de representaciones (cadenas) de todas las fórmulas booleanas que son satisfactibles**. Podría tener la forma: $L(M) = \{\varphi_1, \varphi_2, \varphi_3, \dots\}$, donde φ_i es una fórmula booleana satisfactible.
5. ¿Qué postula la Tesis de Church-Turing?
- La Tesis de Church-Turing postula que **todo dispositivo computacional físicamente realizable puede ser simulado por una Máquina de Turing (MT)**. En otras palabras, la tesis afirma que cualquier proceso que pueda ser considerado como un algoritmo o un método efectivo de cálculo puede ser llevado a cabo por una Máquina de Turing. A pesar de su simplicidad, la Máquina de Turing se considera un modelo computacional suficientemente poderoso como para simular cualquier otra forma concebible de computación.
6. ¿Cuándo dos MT son equivalentes? ¿Y cuándo dos modelos de MT son equivalentes?
- Dos **MT son equivalentes** si aceptan el mismo lenguaje (es decir, si resuelven el mismo problema). **Dos modelos de MT son equivalentes** si dada una MT de un modelo existe una MT equivalente del otro.
-

Ejercicio 2

Dado el alfabeto $\Sigma = \{0, 1\}$:

1. Obtener el conjunto Σ **y el lenguaje incluido en Σ** con cadenas de a lo sumo 2 símbolos.
 - $\Sigma^* = \{\lambda, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, \dots\}$
 - $L = \{\lambda, 0, 1, 00, 01, 10, 11\}$
2. Sea el lenguaje $L = \{0^n 1^n \mid n \geq 0\}$. Obtener los lenguajes $\Sigma \cap L$, $\Sigma \cup L$ y L^C respecto de Σ^* .
 - $\Sigma^* \cap L = L$
 - $\Sigma \cup L = \Sigma$

- $L^{\wedge}C^{\wedge} = \Sigma^* - L$

Ejercicio 3

En clase se mostró una MT no determinística (MTN) que acepta las cadenas de la forma ha^n o hb^n , con $n \geq 0$. Construir (describir la función de transición) una MT determinística (MTD) equivalente.

- **Idea General:**
 - Recorrer la cadena y determinar si después de "h" hay una "a" o una "b".
 - Si hay una "a" luego de "h" buscamos solo "a" recorriendo hacia la derecha, si encontramos una "b" frenamos y retornamos "no".
 - Si hay una "b" luego de "h" buscamos solo "b" recorriendo hacia la derecha, si encontramos una "a" frenamos y retornamos "no".
 - Si recorremos toda la cadena retornamos "si".
- **Estados:** $Q = \{q_0, q_1, q_a, q_b, q_A, q_R\}$
 1. q_0 : Estado inicial.
 2. q_1 : M determina que letra hay.
 3. q_a : M busca una "a".
 4. q_b : M busca una "b".
- **Alfabeto:** $\Gamma = \{h, a, b, B\}$
- **Función de Transición:** δ
 1. $\delta(q_0, h) = \{(q_1, h, R)\} \rightarrow$ Pasamos a determinar que letra hay después.
 2. $\delta(q_1, a) = \{(q_a, a, R)\} \rightarrow$ Encontramos "a" y buscamos más "a".
 3. $\delta(q_1, b) = \{(q_b, b, R)\} \rightarrow$ Encontramos "b" y buscamos más "b".
 4. $\delta(q_1, B) = \{(q_A, B, S)\} \rightarrow$ Encontramos B y aceptamos.
 5. $\delta(q_a, a) = \{(q_a, a, R)\} \rightarrow$ Recorremos siguiendo con las "a".
 6. $\delta(q_a, B) = \{(q_A, B, S)\} \rightarrow$ Llegamos al final y solo había "a".
 7. $\delta(q_b, b) = \{(q_b, b, R)\} \rightarrow$ Recorremos siguiendo con las "b".
 8. $\delta(q_b, B) = \{(q_A, B, S)\} \rightarrow$ Llegamos al final y solo había "b".
- **MT M = (Q, Γ , δ , q_0 , q_A , q_R)**

Ejercicio 4

Describir la idea general de una MT con varias cintas que acepte, de la manera más eficiente posible (menor cantidad de pasos), el lenguaje $L = \{a^n b^n c^n \mid n \geq 0\}$.

- **Idea General:**

- La MT va a utilizar 2 cintas:
 - La primera va a ser utilizada como cinta de entrada de los datos, por ejm. contendría los valores (a, a, a, b, b, b, c, c, c).
 - La segunda va a ser utilizada como una cinta contadora que nos permita llevar un conteo de la cantidad de "a", "b" y "c" que tiene la cadena de la cinta de entrada.
- Si la primera cinta está vacía, es decir, es B, devolvemos "si".
- Si en la primera cinta no encontramos una "a" en primer lugar, devolvemos "no".
- Suponiendo que en la primera cinta nos encontramos con una "a" en primer lugar y la segunda estará B, en ese caso marcamos en la segunda cinta que hay una "a" y avanzamos los 2 cabezales de la cintas hacia la derecha, acá nos podemos encontrar 3 casos:
 - La letra que sigue es una "a", en ese caso marcamos que hay una "a" en la segunda cinta y avanzamos los 2 cabezales hacia la derecha.
 - La letra que sigue es una "b", en ese caso mantenemos el cabezal de la primera cinta en la posición actual (primera "b") y movemos el cabezal de la segunda cinta hacia la izquierda para que quede posicionado en la última "a" contada.
 - La letra que sigue es una "c" o encontramos un B, devolvemos "no".
- Una vez empezamos a contabilizar las "b" recorriendo la primera cinta hacia la derecha y la segunda hacia la izquierda pueden pasar 5 casos:
 - La letra que sigue es una "b" en la primera cinta y en la segunda no hay B, en ese caso movemos el cabezal de la primera cinta hacia la derecha y el de la segunda cinta hacia la izquierda.
 - La letra que sigue es una "c" en la primera cinta y en la segunda hay B, en ese caso mantenemos el cabezal de la primera cinta en la posición actual (primera "c") y movemos el cabezal de la segunda cinta hacia la derecha para que quede posicionado en la primera "a" contada.
 - La letra que sigue es una "a" o encontramos un B en la primera cinta, devolvemos "no".
 - La letra que sigue es una "b" en la primera cinta pero en la segunda hay un B, devolvemos "no" ya que en ese caso la cantidad de "a" y "b" no serían las mismas.
 - La letra que sigue es B en la primera cinta y en la segunda no es B, devolvemos "no".

- Una vez que empezamos a contabilizar las "c" recorriendo las 2 cintas hacia la derecha pueden pasar 5 casos:
 - La letra que sigue es una "c" en la primera cinta y en la segunda no hay B, en ese caso movemos los 2 cabezales hacia la derecha.
 - La letra que sigue es una "a" o una "b" en la primera cinta, devolvemos "no".
 - La letra que sigue es B en la primera cinta y en la segunda no es B, devolvemos "no".
 - La letra que sigue es B en las 2 cintas, devolvemos "si".
 - La letra que sigue es "c" en la primera cinta y en la segunda hay B, devolvemos "no" ya que en ese caso la cantidad de "a", "b" y "c" no serían las mismas.
-

Ejercicio 5

Explicar cómo una MT sin el movimiento S (el no movimiento) puede simular (ejecutar) otra que sí lo tiene.

- Se podría simular haciendo que la MT cuando se llegue por ejm. al elemento en el que se quiera frenar mueva su cabezal hacia la izquierda y luego hacia la derecha, cambiando a un nuevo estado al moverse a la izquierda para garantizar el movimiento a la derecha sin hacer nada.
-

Ejercicio 6

En clase se construyó una MT con 2 cintas que acepta $L = \{w \mid w \in \{a, b\}^* \text{ y } w \text{ es un palíndromo}\}$. *Construir una MT equivalente con 1 cinta.* Ayuda: la solución que vimos para aceptar el lenguaje de las cadenas $a^n b^n$, con $n \geq 1$, puede ser un buen punto de partida.*

- **Idea General:**
 - Recorrer la cadena haciendo una "ida y vuelta" marcando la primera letra y avanzando hacia la derecha hasta la última, si la última letra es igual a la primera que encontramos, la marcamos y retrocedemos hacia la izquierda hasta chocarnos con la siguiente letra sin marcar y repetimos el proceso.
 - Cuando tenemos elementos marcados hacia nuestra derecha y hacia nuestra izquierda eso significa que la cadena es un palíndromo.
- **Estados:** $Q = \{q_0z, q_0d, q_baf, q_bai, q_caf, q_cai, q_bbf, q_bbi, q_cbf, q_cbi, q_A, q_R\}$

1. q_0z : Estado inicial de izquierda a derecha.

2. q_0d : Estado inicial de derecha a izquierda.
3. q_baf : Estado "Buscar A al Final".
4. q_bai : Estado "Buscar A al Inicio".
5. q_caf : Estado "Chequeo de A al Final".
6. q_cai : Estado "Chequeo de A al Inicio".
7. q_bbf : Estado "Buscar B al Final".
8. q_bbi : Estado "Buscar B al Inicio".
9. q_cbf : Estado "Chequeo de B al Final".
10. q_cbi : Estado "Chequeo de B al Inicio".

- **Alfabeto:** $\Gamma = \{a, b, B\}$
- **Función de Transición:** δ

1. $\delta(q_0z, a) = \{(q_baf, B, R)\} \rightarrow$ Iniciamos de izquierda a derecha con "a"
2. $\delta(q_0z, b) = \{(q_bbf, B, R)\} \rightarrow$ Iniciamos de izquierda a derecha con "b"
3. $\delta(q_baf, a) = \{(q_baf, a, R)\} \rightarrow$ Buscamos "a" al final y encontramos "a" en el camino
4. $\delta(q_baf, b) = \{(q_baf, b, R)\} \rightarrow$ Buscamos "a" al final y encontramos "b" en el camino
5. $\delta(q_baf, B) = \{(q_caf, B, L)\} \rightarrow$ Buscamos "a" al final y encontramos B en el camino
6. $\delta(q_caf, a) = \{(q_0d, B, L)\} \rightarrow$ Chequeamos si la letra es una "a" al final
7. $\delta(q_caf, B) = \{(q_A, B, S)\} \rightarrow$ Nos damos cuenta que la cadena es un palíndromo
8. $\delta(q_0d, a) = \{(q_bai, B, L)\} \rightarrow$ Iniciamos de derecha a izquierda con "a"
9. $\delta(q_0d, b) = \{(q_bbi, B, L)\} \rightarrow$ Iniciamos de derecha a izquierda con "b"
10. $\delta(q_bai, a) = \{(q_bai, a, L)\} \rightarrow$ Buscamos "a" al inicio y encontramos "a" en el camino
11. $\delta(q_bai, b) = \{(q_bai, b, L)\} \rightarrow$ Buscamos "b" al inicio y encontramos "b" en el camino
12. $\delta(q_bai, B) = \{(q_cai, B, R)\} \rightarrow$ Buscamos "a" al inicio y encontramos B en el camino
13. $\delta(q_cai, a) = \{(q_0z, B, R)\} \rightarrow$ Chequeamos si la letra es una "a" al inicio
14. $\delta(q_cai, B) = \{(q_A, B, S)\} \rightarrow$ Nos damos cuenta que la cadena es un palíndromo
15. $\delta(q_bbf, a) = \{(q_bbf, a, R)\} \rightarrow$ Buscamos "b" al final y encontramos "a" en el camino
16. $\delta(q_bbf, b) = \{(q_bbf, b, R)\} \rightarrow$ Buscamos "b" al final y encontramos "b" en el camino
17. $\delta(q_bbf, B) = \{(q_cbf, B, L)\} \rightarrow$ Buscamos "b" al final y encontramos B en el camino
18. $\delta(q_cbf, b) = \{(q_0d, B, L)\} \rightarrow$ Chequeamos si la letra es una "b" al final
19. $\delta(q_cbf, B) = \{(q_A, B, S)\} \rightarrow$ Nos damos cuenta que la cadena es un palíndromo
20. $\delta(q_bbi, a) = \{(q_bbi, a, L)\} \rightarrow$ Buscamos "b" al inicio y encontramos "a" en el camino
21. $\delta(q_bbi, b) = \{(q_bbi, b, L)\} \rightarrow$ Buscamos "b" al inicio y encontramos "b" en el camino
22. $\delta(q_bbi, B) = \{(q_cbi, B, R)\} \rightarrow$ Buscamos "b" al inicio y encontramos B en el camino
23. $\delta(q_cbi, b) = \{(q_0z, B, R)\} \rightarrow$ Chequeamos si la letra es una "b" al inicio
24. $\delta(q_cbi, B) = \{(q_A, B, S)\} \rightarrow$ Nos damos cuenta que la cadena es un palíndromo

- **MT M = (Q, Γ , δ , q_0 , q_A , q_R)**
-

Ejercicio 7

Construir una MT que calcule la resta de dos números. *Ayuda: se puede considerar la idea de solución propuesta en clase. ACLARACIÓN: Tomo las precondiciones que se muestran en la clase, los números están codificados en unario, separados por un 0 y las restas dan resultados ≥ 0*

- **Idea General para solución con 2 cintas:**
 - En la primera cinta vamos a tener la resta codificada en unario, por ejm. la resta 4 - 3 sería (11110111).
 - En la segunda cinta lo que vamos a hacer es ir copiando hacia la derecha el primer número a restar y cuando lleguemos al inicio del segundo número iremos tachando hacia la izquierda el primer número copiado, lo que quede en la segunda cinta será el resultado a devolver.
 - **Estados:** $Q = \{q_0, q_{copiar}, q_{restar}, q_A, q_R\}$
 1. q_0 : Estado inicial.
 2. q_{copiar} : Estado para copiar el primer número.
 3. q_{restar} : Estado para ir restando en la segunda cinta.
 - **Alfabeto:** $\Gamma = \{0, 1, B\}$
 - **Función de Transición:** δ
 1. $\delta(q_0, 1, B) = \{(q_{copiar}, (1, R), (1, R))\} \rightarrow$ Empezamos a copiar el primer número
 2. $\delta(q_{copiar}, 1, B) = \{(q_{copiar}, (1, R), (1, R))\} \rightarrow$ Recorremos el primer número y copiamos
 3. $\delta(q_{copiar}, 0, B) = \{(q_{restar}, (0, R), (B, L))\} \rightarrow$ Llegamos al final del primer número
 4. $\delta(q_{restar}, 1, 1) = \{(q_{restar}, (1, R), (0, L))\} \rightarrow$ Tachamos en la segunda cinta
 5. $\delta(q_{restar}, B, 1) = \{(q_A, (B, S), (1, S))\} \rightarrow$ Terminamos el segundo número y este era más chico
 6. $\delta(q_{restar}, B, B) = \{(q_A, (B, S), (B, S))\} \rightarrow$ Terminamos el segundo número y la resta es 0
 - **MT M = (Q, Γ , δ , q_0 , q_A , q_R)**
-

Ejercicio 8

Construir una MT que genere todas las cadenas de la forma $a^n b^n$, con $n \geq 1$. *Ayuda: se puede considerar la idea de solución propuesta en clase.*

- **Idea General para solución con 2 cintas:**

- En la primera cinta vamos a tener todas las cadenas (ab, aabb, aaabbb, aaaabbbb, ...).
- La segunda cinta la vamos a usar como un contador para ir viendo la cantidad de "a" y de "b" que tenemos que ir poniendo en la primera cinta.

- **Estados:** $Q = \{q_0, q_a, q_b, q_A, q_R\}$

1. q_0 : Estado inicial.
2. q_a : Estado para escribir una "a".
3. q_b : Estado para escribir una "b" y delimitar con ",".

- **Alfabeto:** $\Gamma = \{a, b, ", ", B\}$

- **Función de Transición:** δ

1. $\delta(q_0, B, B) = \{(q_a, (B, S), (1, S))\} \rightarrow$ Preparamos el contador de la segunda cinta
2. $\delta(q_a, B, 1) = \{(q_a, (a, R), (1, R))\} \rightarrow$ Copiamos una "a" en la primera cinta
3. $\delta(q_a, B, B) = \{(q_b, (B, S), (B, L))\} \rightarrow$ Preparamos las cintas para copiar las "b"
4. $\delta(q_b, B, 1) = \{(q_b, (b, R), (1, L))\} \rightarrow$ Copiamos una "b" en la primera cinta
5. $\delta(q_b, B, B) = \{(q_a, (", ", R), (1, S))\} \rightarrow$ Delimitamos el fin de la cadena $a^n b^n$ y preparamos las cintas para copiar las "a"

- **MT $M = (Q, \Gamma, \delta, q_0, q_A, q_R)$**