Explicación Práctica

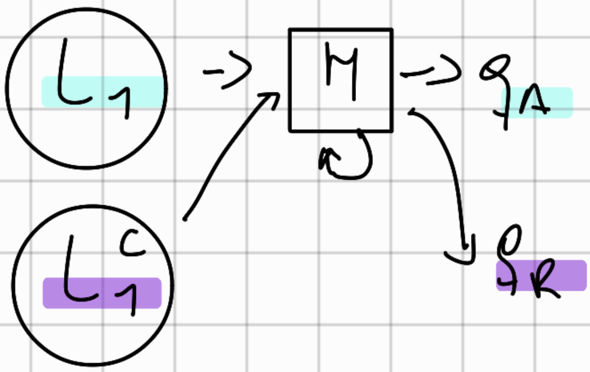
↳ MT: para una entrada devuelve un resultado. Modelo simple de computadora.

Hay 2 problemas, de búsqueda y de decisión.
un camino o cosas así — sí o no

Un lenguaje es un conjunto de cadenas / strings. Los lenguajes son conjuntos y aceptan op. de conjuntos. El lenguaje de una máquina de Turing es aquel que la máquina acepta.
El lenguaje asociado a un problema es aquel string que representa ese problema.

Las MT suelen tener una cinta → si tengo tal estado en la cinta luego algo, esto se puede aplicar en N cintas (N → nº finito). + cintas ≠ + pot.

En el cuadrito suele estar escrita todas las op. y al costado los estados (los representan con q_{algo}), en las intersecciones se dice el estado y lo que hace esa cinta.



$L(M) = L_1$ → es cuando me da q_A.

Σ^* → sigma estrella
↘ conjunto de todos los strings.

↳ MT no determinística: ante una misma entrada puede tomar dos caminos o + a la vez.

· MT reconocedora: devuelve q_A o q_R

· MT calculadora: devuelve una cadena.

· MT generadora: genera un lenguaje a partir de una cinta vacía.

Práctica 1 - FTC

Ejercicio 1

1. La diferencia entre los problemas de búsqueda y de decisión es que los de decisión reciben un problema y la respuesta sólo puede ser "sí" o "no". Un problema de búsqueda puede devolver múltiples respuestas.

2. Los lenguajes son la cadena de símbolos, una MT acepta ese lenguaje que representa algo (por ej. el problema). Si ese lenguaje se lo doy a la MT me da q_A.

3. "obtener un conjunto de asignaciones que hace verdadera a φ.

4. $L = \{ L \mid L$ es una formula booleana válida.

5. La tesis de Church-Turing postula que todo lo computable puede ser computado por una máquina de turing.

6. Dos MT son equivalentes cuando aceptan el mismo lenguaje.
   Dos modelos de MT son equivalentes si dado una MT de un modelo existe una MT equivalente del otro.

Ejercicio 2

$\Sigma = \{0, 1\}$

① $\Sigma^* = \{\lambda, 0, 1, 00, 01, 10, 11, 001, 010, \dots\}$

$L = \{\lambda, 0, 1, 00, 01, 10, 11\}$

② $L = \{0^n 1^n \mid n \geq 0\}$

- $\Sigma^* \cap L = L$, $L \subseteq \Sigma^*$
- $\Sigma^* \cup L = \Sigma^*$, $L \subseteq \Sigma^*$
- $L^c = \Sigma^* - L$

Ejercicio 3

- MTN: no devuelve siempre lo mismo
- MTD: siempre devuelve el mismo resultado para una entrada en particular.

Datos comunes:

$M = \{Q, \Sigma, \delta, q_0, q_A, q_R\}$

$Q = \{q_0, q_A, q_R, q_h, q_a, q_b\}$

$\Sigma = \{h, a, b\}$

Estados:

- q_0: estado inicial
- q_h: M está en "h"
- q_a: M está en "a"
- q_b: M está en "b"

Matriz

	h	a	b	B
q_0	q_h, h, R			
q_h		q_a, a, R	q_b, b, R	
q_a		q_a, a, R		q_A, B, S
q_b			q_b, b, R	q_A, B, S

Ejercicio 4

$$L = \{ a^n b^n c^n \mid n \geq 0 \}$$

Idea general:

1° Copia todos los símbolos "a" a la cinta 2. El cabezal termina cuando encuentra una "b" en la cinta 1.

2° En cada iteración me muevo a la derecha en la cinta 1, recorriendo las "b", y a la izquierda de la cinta 2, sobreescribiendo las "a" por "b".

3° En cada iteración, me muevo a la derecha de la cinta 1, leyendo las "c", y a la derecha de la cinta 2, sobreescribiendo las "b" por "c".

4° Se acepta la entrada sí y sólo si ambos cursores están en B.

$M = \{Q, \Sigma, \delta, q_0, q_A, q_R\}$

$Q = \{q_0, q_a, q_b, q_c, q_A, q_R\}$

$\Sigma = \{a, b, c, B\}$

q_0: estado inicial
q_a: copiando "a" en cinta 2
q_b: copiando "b" en cinta 2
q_c: copiando "c" en cinta 2

Transiciones:

① $\delta(q_0, (a, B)) = (q_a, (a, R), (a, R))$

② $\delta(q_2, (a, B)) = (q_2, (a, R), (a, R))$

③ $\delta(q_2, (b, B)) = (q_b, (b, R), (B, L))$

④ $\delta(q_b, (b, B)) = (q_b, (b, R), (b, L))$

⑤ $\delta(q_b, (c, B)) = (q_b, (c, \underline{B}), (B, R))$

⑥ $\delta(q_c, (c, B)) = (q_c, (c, R), (c, R))$

⑦ $\delta(q_c, (c, B)) = (q_A, (c, R), (B, R))$