Práctica 3

① $L_U = \{(\langle M \rangle, w) \mid M \text{ acepta } w\}$

Se quiere demostrar que $L_U \in RE$.
Se construye una MT M_U que determine si w está en L_U.
Como no sabemos el comportamiento de $\langle M \rangle$ cuando w no pertenece a L_U, no podemos asegurar que se detendrá y por lo tanto M_U tampoco.
Por eso, M_U puede ejecutar $\langle M \rangle$:

- Si $\langle M \rangle$ termina en q_A, M_U termina en q_A
- Si $\langle M \rangle$ termina en q_R, M_U termina en q_R
- Sino, M_U puede quedar loopeando

$L_{HP} = \{(\langle M \rangle, w) \mid M \text{ para a partir de } w\}$

Se quiere demostrar que $L_{HP} \in RE$
Se construirá una MT M_{HP} que ejecute $\langle M \rangle$ a partir de w.
Como $\langle M \rangle$ se ejecuta sobre M_{HP}, sabemos que va a parar, aceptando o rechazando, a partir de w. No podemos garantizar que $\langle M \rangle$ pare, por lo tanto no podemos garantizarlo de M_{HP}.
M_{HP} deberá finalizar con q_A independientemente de si $\langle M \rangle$ terminó en q_R o q_A.
Al ejecutar M_{HP} con $\langle M \rangle$:

- Si $\langle M \rangle$ termina en q_A, M_{HP} termina en q_A
- Si $\langle M \rangle$ termina en q_R, M_{HP} termina en q_A
- Sino, M_{HP} puede quedar loopeando

celdas / # estados / # símbolos

$(N.Q.\Sigma)^N$

(2)

A. Sabemos que M debe parar antes de cierta cantidad de pasos que lo hagan entrar en loop.

B. Si asumimos que $M \in R$, entonces sabemos que M parará cuando lea un símbolo que no pertenece a $L(M)$.

C. Si asumimos que $M \in R$, entonces sabemos que M parará cuando lea un símbolo que no pertenece a $L(M)$.
Para que se ejecuten 10 pasos necesitamos 10 símbolos.
Para garantizar que llegó a 10 se puede usar un contador en otra cinta o bien usar una MT_v que restrinja la cantidad de celdas. CONSULTAR!!

D. No se puede garantizar que luego de los 10 símbolos, M pare.

(3) Se usaría notación unaria:

Números pares:

1) Cinta 1 y 2 en blanco.
2) Marco con un 0 el comienzo de un número en la cinta 2.
3) En la cinta 2 escribo 1s por cada 1s de la cinta 1.
4) En la cinta 1 agrego 2 1s.
5) Volver al punto 2).

Números enteros:

1) Cinta 1, 2 y 3 en blanco.
2) Marco con un 0 el comienzo de un número en la cinta 2.
3) Marco con dos 0 el comienzo de un numero negativo en cinta 3.
4) En la cinta 2 y 3 escribo 1s por cada 1s de la cinta 1.
5) En la cinta 1 agrego 1 1s.
6) Volver al punto 2).

Números racionales

1)

Cadenas de Σ

1) Cinta 1 y 2 en blanco.
2) Marco con un " " el comienzo de un número en la cinta 2.
3) En la cinta 2 escribo los símbolos de la 1.
4) En la cinta 1: si hay un 0 escribo un 1 y si hay un 1 escribo 0.
5) Vuelvo al 2.

(4) Se asume que f_{HP} es total computable ya que siempre finaliza.
Se supone que $L(f_{HP}) \in R$.
Por definición, f_{HP} devuelve 0 si M no para a partir de w o bien $x \neq (\langle M \rangle, w)$.
Se espera que f_{HP} devuelva un resultado incluso si entra en bucle, M_{HP} "siempre para".
A esto se lo conoce como Halting Problem (HP),

que pertenece a RE.

Al decir que $HP \in R$ se llega a una contradicción.

Por lo tanto, es falso que f_{HP} es total computable.

5