CyC - Practica 5

Facundo Tomatis

(Ejercicio 1)

Consigna:
Sean L_1 y L_2, dos lenguajes definidos sobre {0,1}*

- $L_1 = \{0^n 1 / n \geq 0\}$

- $L_2 = \{1^n 0 / n \geq 0\}$

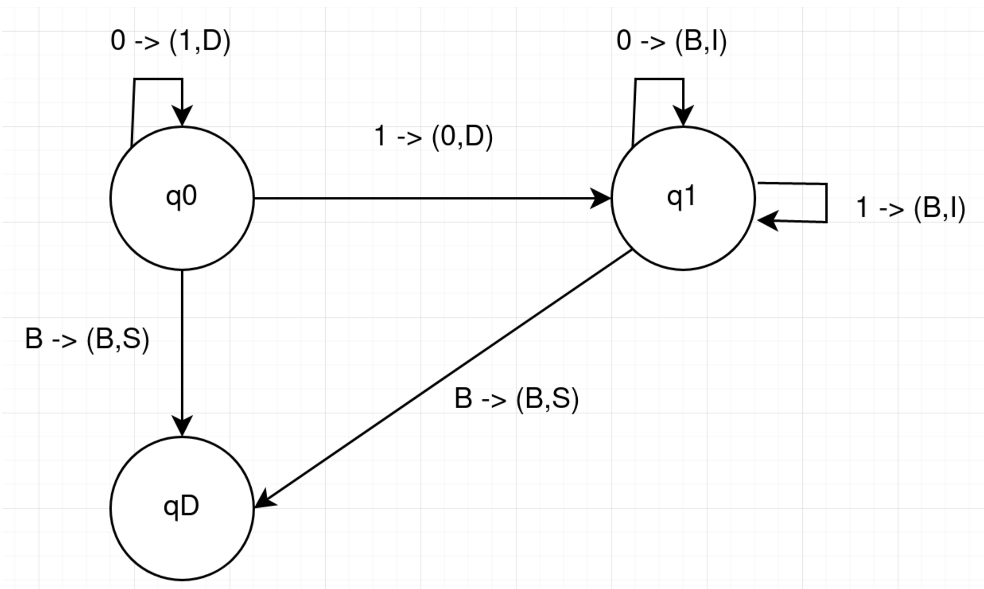
a) Demuestre que existe una reducción ($L_1 \ \alpha \ L_2$)

b) Idem para $L_2 = \{\lambda\}$

c) Idem para $L_2 = \{1^n 0 / n > 0\}$

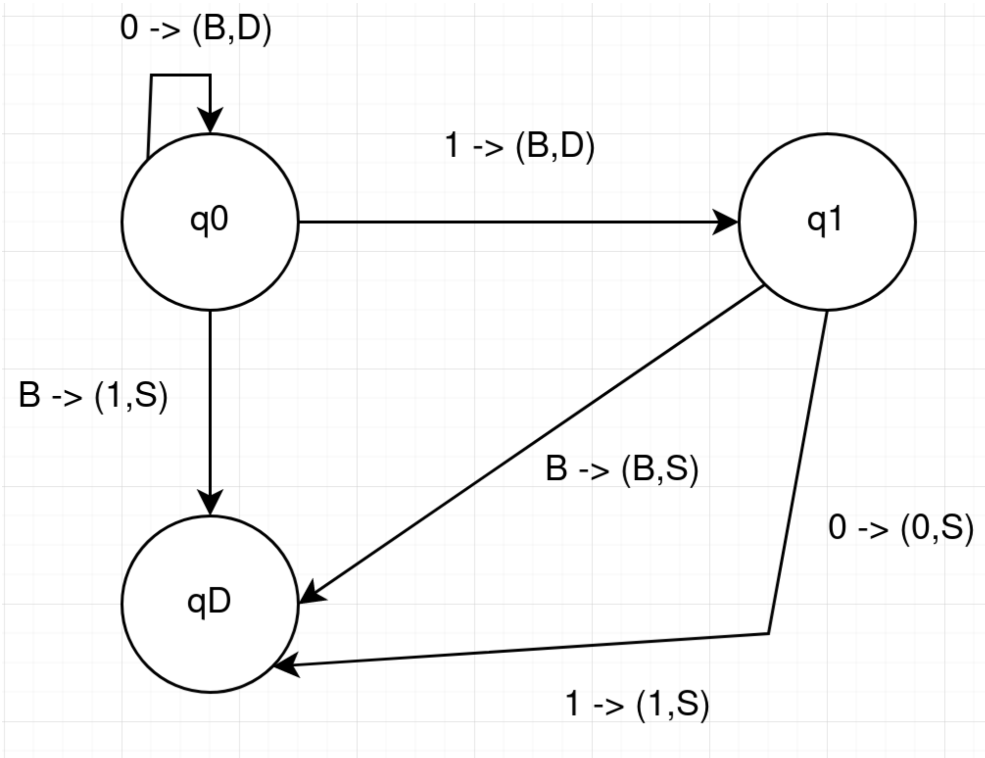
Respuesta:

a) $L_1 \ \alpha \ L_2$
M_f es una maquina de turing que ejecuta la funcion de reducibilidad.
M_f=$<Q, \Sigma, \Gamma, \delta, q_0, q_d>$ $Q = \{q_0\}$ $\Sigma = \{0, 1\}$ $\Gamma = \{B, 0, 1\}$

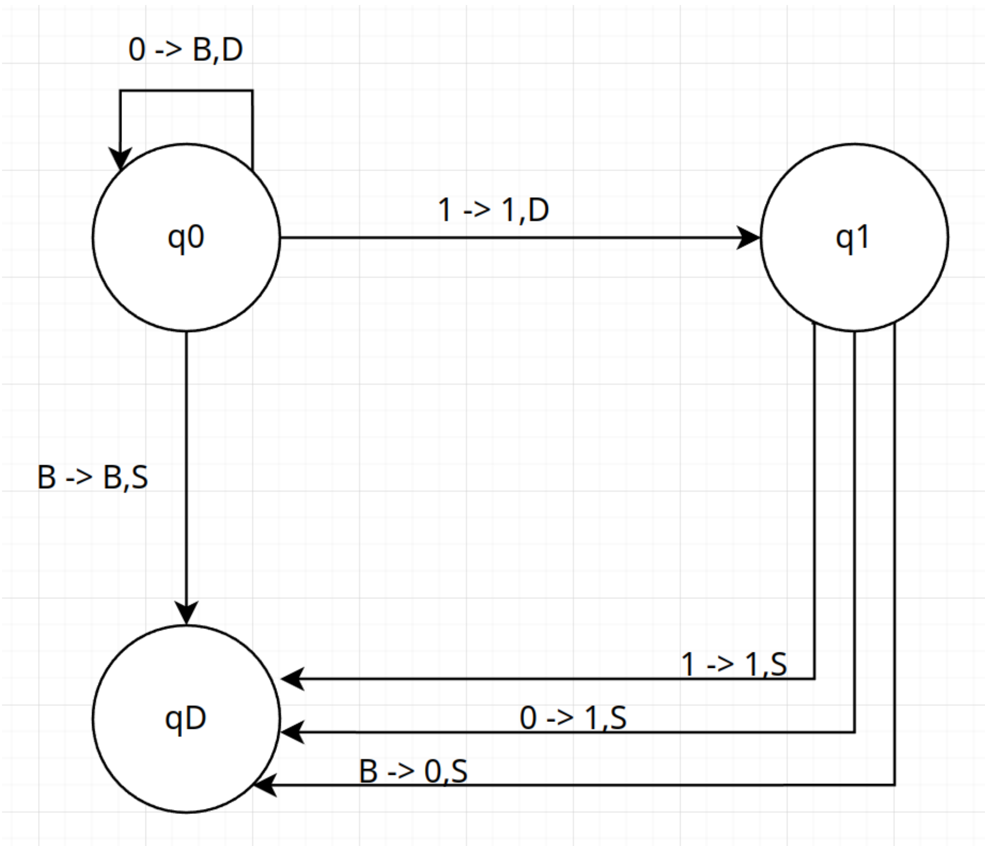


1. M_f siempre se detiene ☑ ya que el alfabeto es finito.
2. $w \in L_1 \Leftrightarrow M_f(w) \in L_2$ ☑ se puede observar viendo la MT

b) con $L_2 = \{\lambda\}$



c) con $L_2 = \{1^n 0 / n > 0\}$



(Ejercicio 2)

Consigna:
Sean L_1 y L_2, dos lenguajes tales que existe una reducción ($L_1 \propto L_2$)

a) Qué se puede afirmar de L_1 si se sabe que $L_2 \in$ R

b) Qué se puede afirmar de L_1 si se sabe que $L_2 \in$ (CO-RE $-$ RE)

c) Qué se puede afirmar de L_2 si se sabe que $L_1 \in$ R

d) Qué se puede afirmar de L_2 si se sabe que $L_1 \in$ (CO-RE $-$ RE)

Respuesta:

a) Se puede afirmar que si $L_2 \in$ R $\Rightarrow L_1 \in$ R. (demostracion en diapos)

b) Se puede afirmar que $L_1 \in$ CO-RE, tambien podemos saber que $\overline{L_1} \in RE$ ya que $\overline{L_2} \in RE$ por lo que $L_1 \notin RE$, entonces se puede afirmar que $L_1 \in$ (CO-RE $-RE$)

c) Nada, ya que este puede ser un lenguaje de cualquiera de $\mathscr{L}$

d) Se puede afirmar que: (CO-RE $-$ RE) $=$ (CO-RE $-$ R) $\therefore L_1 \notin R, L_1 \notin RE \Rightarrow L_2 \notin R, L_2 \notin RE$ por contrarreciproca

---

(Ejercicio 3)

Consigna:
Sean HP y L_u los lenguajes *Halting Problem* y *Lenguaje Universal* respectivamente.

- HP $= \{$(<M>,w) / M se detiene con input $w\}$
- $L_u = \{$(<M>,w) / M acepta $w\}$

Demuestre que existe una reducción HP $\propto L_u$

Respuesta:
HP se reduce a L_u construyendo una maquina M_f que si la maquina es invalida (en HP acepta esto) hacer una maquina <M'> que acepte todo w, si el par es invalido la maquina queda igual, si la maquina rechaza w (transicion a q_R que en HP luego va a q_A) cambiar la delta de transicion q_R por q_A, estos cambios son finitos y el checkeo de si la maquina es invalida o no solo requiere utilizar un parser que tiene finitos movimientos.

(Ejercicio 4)

Consigna:
Sea HP_λ el problema de detención a partir de la cinta en blanco

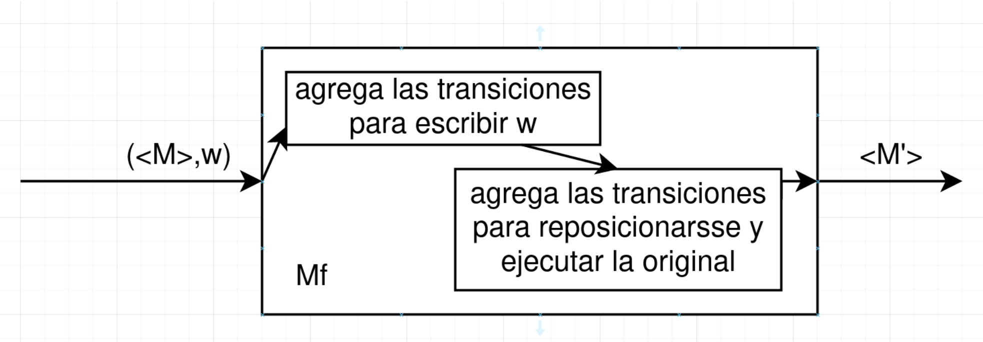
- HP_λ =$\{$<M> / M se detiene con input $\lambda\}$

Demuestre que existe una reducción $HP \ \alpha \ HP_\lambda$

Respuesta:

Existe una M_f que tenga como entrada la tupla (<M>, w) y tenga como salida una <M'> tal que esta borre la cinta y una vez borrada escribe la palabra w "a mano", luego vaya a la posición inicial de la palabra y comience a funcionar como la maquina original. Por lo que si <M> paraba con w <M'> va a parar y si <M> loopeaba con w <M'> tambien va a loopear.

Esta M_f es computable ya que realiza finitas modificaciones a la maquina original (la palabra es finita y por lo tanto el reposicionarse tambien).



(Ejercicio 5)

Consigna:

Demuestre que $L_V \notin$ RE

- $L_V = \{(\text{<M>})/\text{L(M)} = \varnothing\}$

Considere que si <M> es un código inválido de máquina de Turing también pertenece a L_V (ya que no reconoce ningún string). Así L_V es el complemento del lenguaje $L_{NV} = \{(\text{<M>})/\text{L(M)} \neq \varnothing\}$

(*Ayuda:* puede utilizar el complemento de L_u para encontrar una reducción)

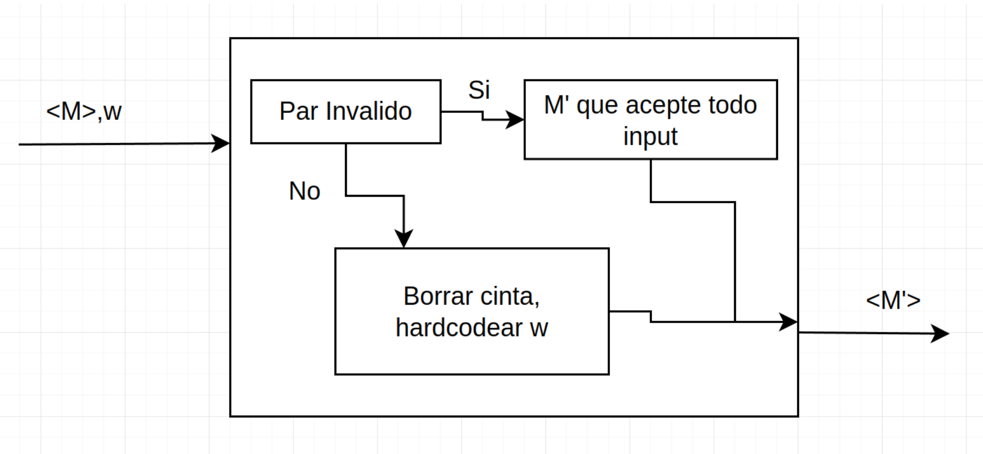
Respuesta:

Ver ejercicio 13 practica 4:

$L_{NV} = \overline{L_V} \implies L_{NV} \in RE, \overline{L_V} \in RE \implies L_V \notin RE$

Otra forma:

$\overline{L_u} \ \alpha \ L_V$



M_f de $\overline{L_u} \ \alpha \ L_V$