UNIVERSIDAD NACIONAL DE LA PLATA

CyC - Practica 4

Facundo Tomatis

(Ejercicio 1)

Consigna:
Construir una máquina de Turing que escriba en la primera cinta las palabras de $\{0,1\}^*$ en orden canónico separadas por un símbolo ";". Obviamente esta máquina nunca se detiene.
Respuesta:

SimuladorMT

```
q0,B,B
q1,;,D,1,S

q1,B,1
q2,B,S,0,I

q1,B,0
q3,B,S,1,I

q2,B,1
q2,B,S,0,I

q2,B,0
q3,B,S,1,I

q2,B,B
q4,B,S,1,D
```

```
q3,B,0
q3,B,S,0,I

q3,B,1
q3,B,S,1,I

q3,B,B
q5,B,S,B,D

q5,B,1
q4,B,S,1,D

q4,B,0
q4,0,D,0,D

q4,B,1
q4,1,D,1,D

q4,B,B
q1,;,D,B,I
```

(Ejercicio 2)

Consigna:
Sean $\Sigma = \{a, b\}$ y $\mathscr{L}$ el conjunto de todos los lenguajes definidos sobre Σ. Diga si las siguientes afirmaciones son verdaderas o falsas:

1.$\mathscr{L}-$R$= \varnothing$	2. $\Sigma^* \in$ R	3. ab $\in \Sigma^*$
4. RE - R $\neq \varnothing$	5. $\varnothing \in$ RE	6. CO-R $\subset$ CO-RE
7. $\{\lambda\} \in (\mathscr{L}-$CO-RE)	8. CO-RE = RE	9. a $\in$ R
10. RE $\cup$ R $= \mathscr{L}$	11.($\mathscr{L}-$RE) =CO-RE	12.{a} $\in$ RE

Respuesta:

1. $\mathscr{L}-$R$= \varnothing$ ✗ Contraejemplo: L_D no esta en R y esta en $\mathscr{L}$
2. $\Sigma^* \in$ R ☑ Existe una MT tal que acepta el lenguaje Σ^* $(q_0 w \vdash q_A)$
3. ab $\in \Sigma^*$ ☑ $\Sigma^* = \{a, b, aa, ab, ...\} \therefore ab \in \Sigma^*$
4. RE - R $\neq \varnothing$ ☑ Por absurdo: $RE - R = \varnothing$ y $L_U \in RE - R$ es absurdo, $\therefore RE - R \neq \varnothing$

5. $\varnothing \in$ RE ☑ Existe una MT que acepta el lenguaje $\varnothing$ tal que $(q_0 w \vdash q_R) \therefore \varnothing \in R$ y por definicion $\varnothing \in RE$

6. CO-R $\subset$ CO-RE ☑ $L \in$ CO-R$\Leftrightarrow \overline{L} \in R \Leftrightarrow \overline{L} \in RE \Leftrightarrow L \in$ CO-RE

7. $\{\lambda\} \in (\mathscr{L}-$CO-RE) ☒ Existe una MT tal que acepte el lenguaje $\{\lambda\}$ y sea recursivo $(q_0 w \vdash q_A, w = \lambda)$ c.c $(q_0 w \vdash q_R) \therefore \{\lambda\} \notin (\mathscr{L}-$ CO-RE)

8. CO-RE = RE ☒ Por absurdo: CO-RE - RE=$\varnothing$ y existe $L_D \in$ (CO-RE - RE) $\therefore$ CO-RE - RE $\neq \varnothing$

9. a $\in$ R ☒ R={$\varnothing$, {a},{b},{a,b},...} $\therefore$ a $\notin$ R

10. RE $\cup$ R = $\mathscr{L}$ ☒ $RE \cup R = RE$ y existe $L_D \notin RE, L_D \in \mathscr{L} \therefore RE \cup R \neq \mathscr{L}$

11. {a} $\in$ RE ☑ {a} $\in$ R $\Rightarrow$ {a} $\in$ RE

12. ($\mathscr{L}-$RE) =CO-RE ☒ ($\mathscr{L}-$RE) = CO-RE $\Leftrightarrow \mathscr{L}-RE-$CO-RE = $\varnothing$
y existe un lenguaje $L = \{1w/w \in L_D\} \cup \{0w/w \notin L_D\}$ tal que L $\notin$ (RE $\cup$ CO-RE) $\therefore$ ($\mathscr{L}-$RE) $\neq$ CO-RE

(Ejercicio 3)

Consigna:
Si L$\in$(RE - R)

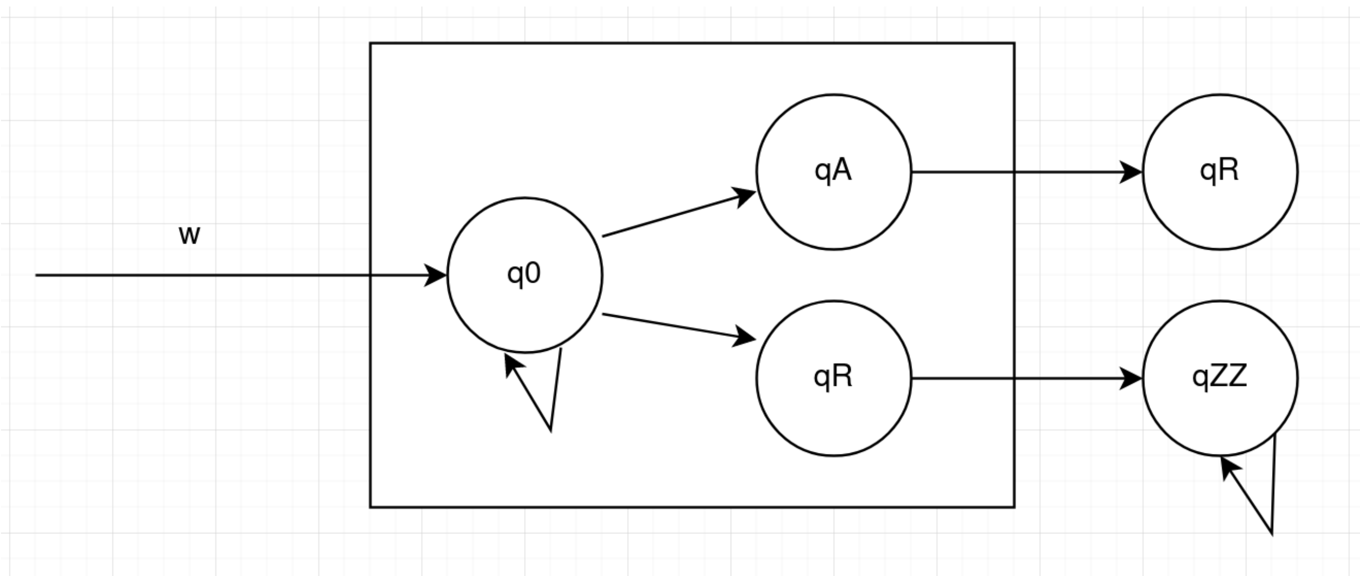
a) ¿Existirá alguna máquina de Turing que rechace parando en q_R si su entrada está en L y rechace loopeando si su entrada no está en L?

b) ¿Existirá alguna máquina de Turing que rechace loopeando si su entrada está en L y rechace parando en q_R si su entrada no está en L?

c) De existir, que lenguaje reconocería esta máquina de Turing.

Respuesta:

a) .



b) No, ya que no puedo saber si va a parar o seguir loopeando si la entrada no esta en L

c) La primer maquina aceptaria el lenguaje $\varnothing$ ya que rechaza cualquier sea la entrada.

(Ejercicio 4)

Consigna:
Sea L $=\{w|$ Existe alguna Máquina de Turing M que acepta w$\}$
¿L $\in$ R? Justifique.
Respuesta:
Si lo acepta significa que el lenguaje es RE, no se detalla que va a parar por lo tanto no estoy seguro de que sea recursivo.

(Ejercicio 5)

Consigna:
Conteste y justifique:

a) ¿$\mathscr{L}$ es un conjunto infinito contable?

b) ¿RE es un conjunto infinito contable?

c) ¿$\mathscr{L}$ − RE es un conjunto infinito contable?

d) Existe algún lenguaje L $\in \mathscr{L}$, tal que L sea infinito no contable

Respuesta:

a) No, ya que $\mathscr{L}$ es lo mismo que decir $\rho(\Sigma^*)$, se demostro anteriormente el teorema que demuestra la siguiente afirmación para cualquier conjunto: $|A| < |\rho(A)|$, donde $\rho(A)$ es no contable.

b) Si, ya que se puede mapear la codificacion de una maquina de turing a un numero natural.

c) No ya que $\mathscr{L}$ es infinito incontable y la resta de un conjunto infinito incontable con un infinito contable es infinito incontable.

d) No, si es un lenguaje con un alfabeto Σ siempre va a ser contable.

(Ejercicio 6)

Consigna:
Sea L un lenguaje definido sobre Σ. Demostrar que:

a) $\overline{L} \notin R \Rightarrow L \notin R$

b) ($L_1 \in$ RE) AND ($L_2 \in$ RE) $\Rightarrow L_1 \cap L_2 \in$ RE

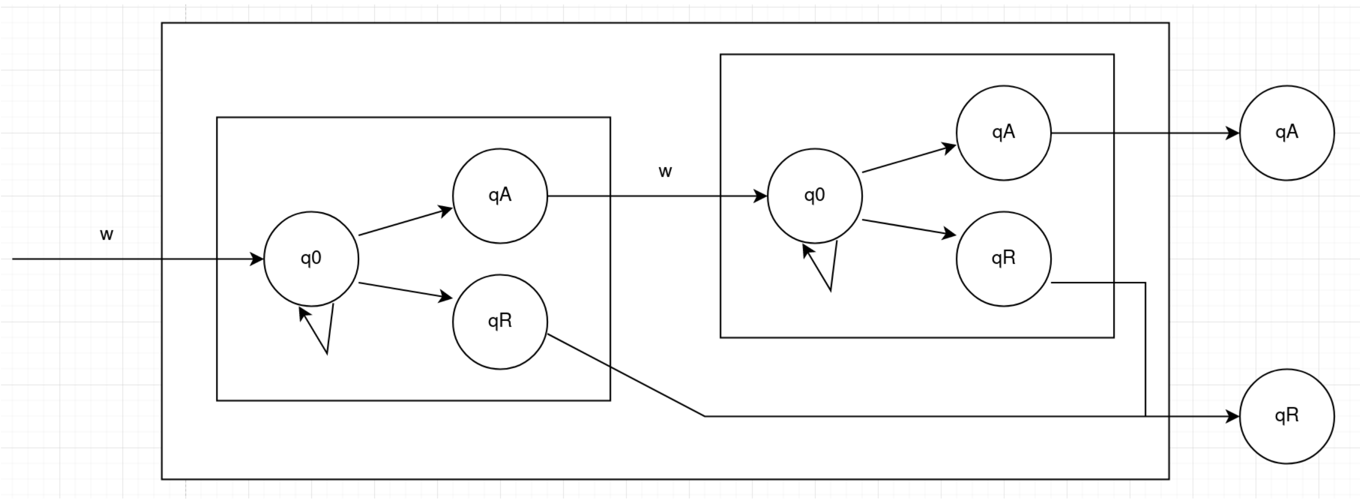
c) ($L_1 \in$ RE) AND ($L_2 \in$ RE) $\Rightarrow L_1 \cup L_2 \in$ RE

d) La unión de un número finito de lenguajes recursivamente enumerables es un lenguaje recursivamente enumerable.

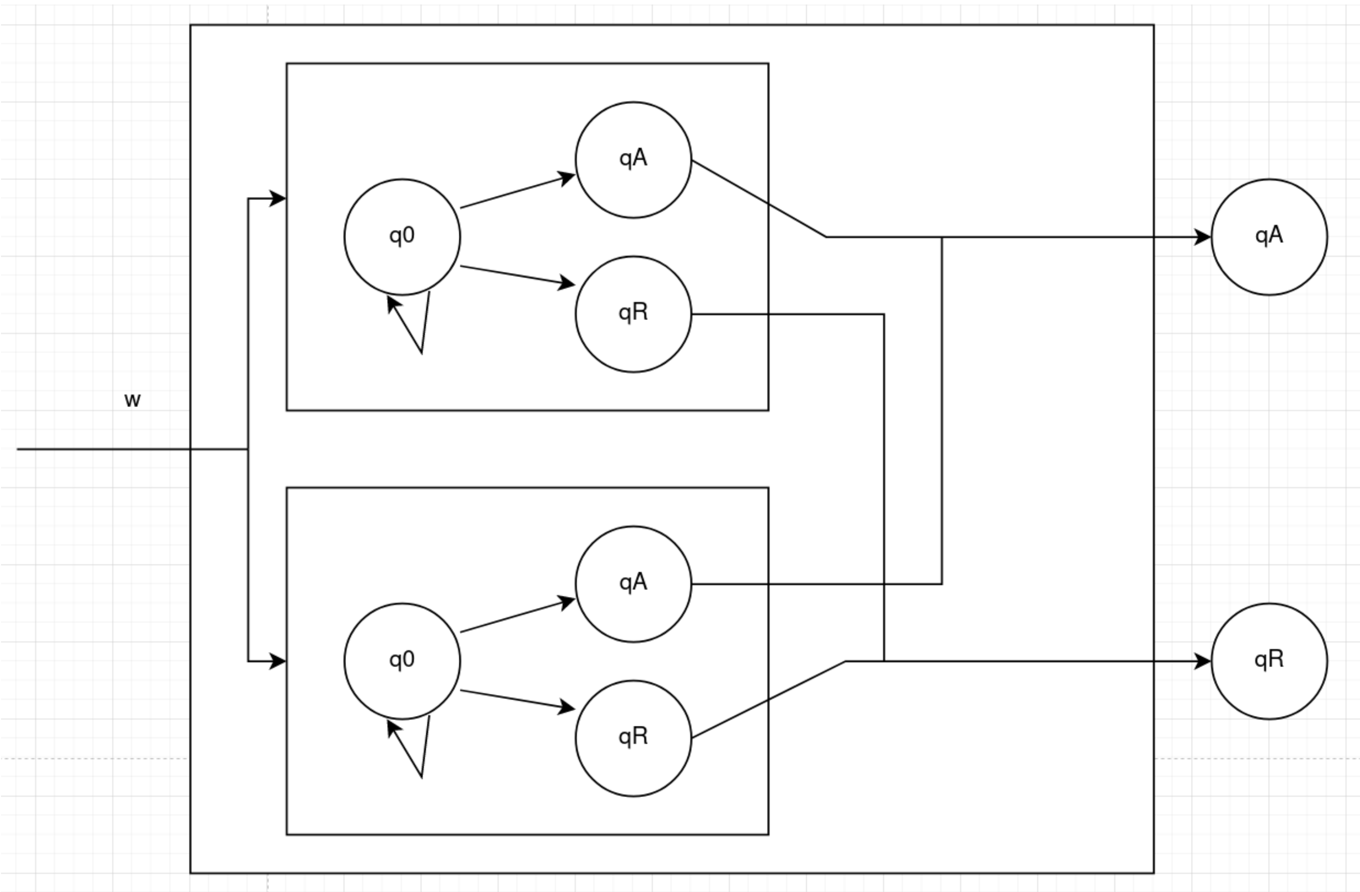
Respuesta:

a) $\overline{L} \notin R \Leftrightarrow L \notin$ CO-R $\Leftrightarrow L \notin R$ ya que CO-R=R

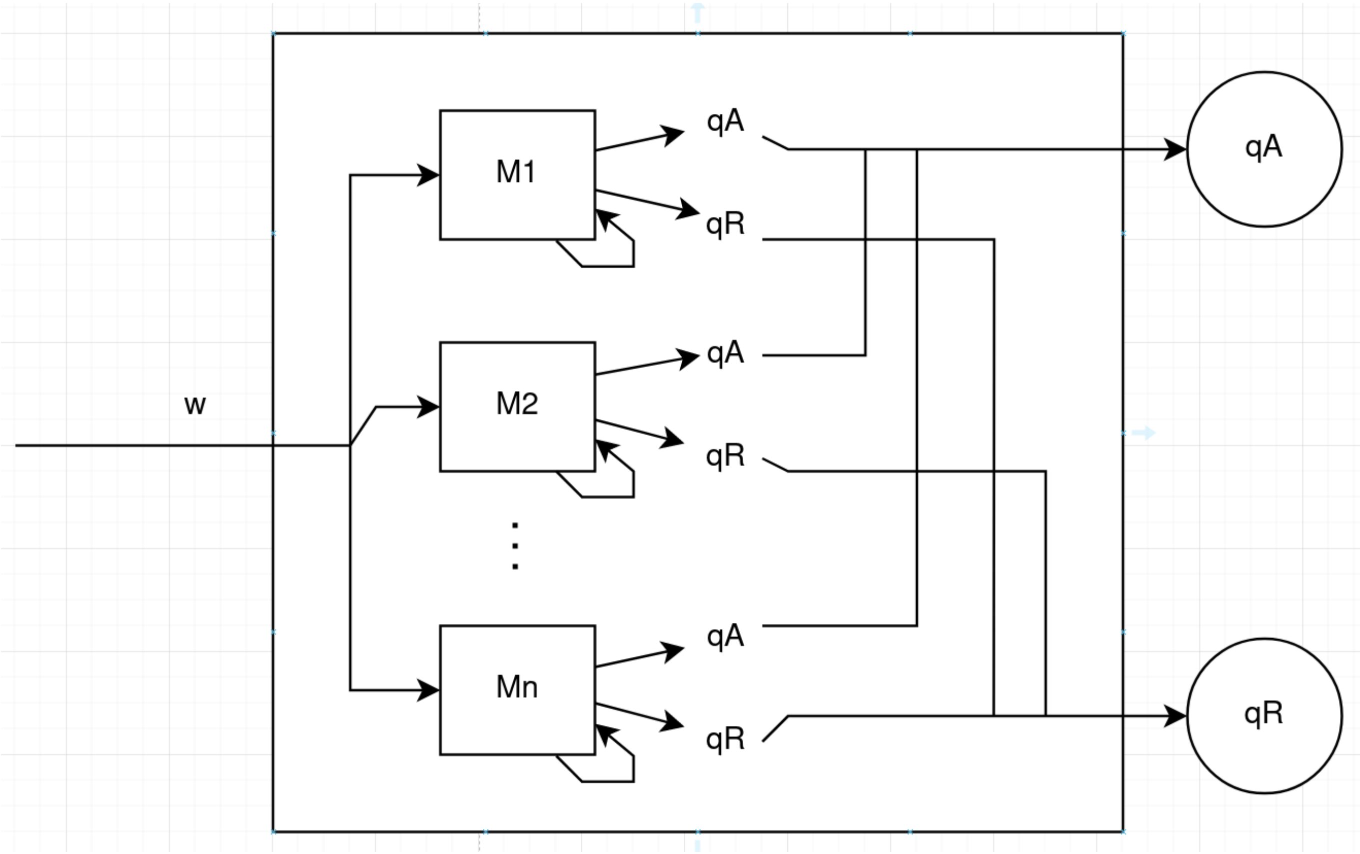
b) .



c) .



d) .



(Ejercicio 7)

Consigna:
Para los casos a), b) y c) del punto anterior ¿valen las recíprocas? Justifique.
Respuesta:

a) $L \notin R \Leftrightarrow L \notin$ CO-R $\Leftrightarrow \overline{L} \notin R$ ya que CO-R=R

b) No, ya que por ejemplo $L_D \cap \overline{L_D} = \varnothing$ y $\varnothing \in$ RE. $L_D \in$ CO-RE, $\overline{L_D} \in$ RE

c) No, ya que por ejemplo $L_D \cup \overline{L_D} = \Sigma^*$ y $\Sigma^* \in$ RE.

(Ejercicio 8)

Consigna:
Si L es un subconjunto de un lenguaje recursivamente enumerable, ¿Puede afirmarse entonces que L es recursivamente enumerable? Justifique.
Respuesta:
Si L es un subconjunto de un lenguaje recursivamente enumerable va a ser recursivamente enumerable solamente que va a ser aceptado menos veces. **Esto puede ser que este mal, que pasa si L es subconjunto de Σ^* y que el mismo no pertenezca a RE**

(Ejercicio 9)

Consigna:
Dado L_1, un lenguaje recursivo cualquiera
$L_2 = \{$<M>| L(M)= $L_1\}$
$L_3 = \{$<M>| L(M)= L_1 y M siempre se detiene$\}$
Determine si $(L_2 - L_3) = \varnothing$. Justifique su respuesta.
Respuesta:
Contraejemplo: Existe una maquina <M> que cuando recibe cualquier w lo rechaza loopeando en el estado inicial $[(q_0, x) \rightarrow (q_0, x, S)$ tal que $x \in \Gamma]$, el lenguaje de esta maquina es $\varnothing$ que pertenece al conjunto de lenguajes R y la codificacion de esta maquina perteneceria al conjunto L_2 ya que lo acepta pero no pertenece al conjunto L_3 ya que nunca se detiene $\therefore L_2 - L_3 \neq \varnothing$

(Ejercicio 10)

Consigna:
Sean los lenguajes $L = \{$<M>| M siempre se detiene$\}$ y $L_R = \{$<M>| L(M) $\in R\}$.
Cuál es la afirmación correcta:

a) $L \subset L_R$

b) $L \supset L_R$

c) $L = L_R$

Respuesta:
Al igual que el ejercicio 9 existe una maquina que puede rechazar loopeando si acepta un lenguaje recursivo por lo que la opción correcta es $L \subset L_R$

(Ejercicio 11)

Consigna:
Encuentre una justificación para cada una de las siguientes afirmaciones

a) $\varnothing \in$ RE

b) Si L es un lenguaje formado por una sola palabra, entonces L $\in$ R

c) Si L es un lenguaje finito, entonces L $\in$ R

Respuesta:

a) $\varnothing$ es un lenguaje, $\varnothing \in$ R y por definición $\varnothing \in$ RE

b) Ejemplo, existe una MT tal que ($q_0 w \vdash q_A$, si w pertenece al lenguaje, c.c q_R)

c) Es el mismo ejemplo, si w pertenece al lenguaje aceptarlo, si no rechazarlo

---

(Ejercicio 12)

Consigna:
Demuestre que si el Halting Problem (HP) es un lenguaje recursivo entonces podría construirse una máquina de Turing que acepte el lenguaje universal L_u, y que se detenga para todo $w \in \Sigma^*$ ¿Qué puede decir entonces sobre la recursividad de HP?
L_u = {(<M>, w) / M acepta w}
HP = {(<M>, w) / M se detiene con input w}
Respuesta:
Se puede hacer la reduccion $L_u \ \alpha \ HP$ y como $L_u \notin R$ poor lo tanto $HP \notin R$
Demostracion:
MT llamada M_f que computa la funcion de reducibilidad f
$M_f((\text{<M>}, w)) = (\text{<M'>}, w)$
Si el par no es valido o la maquina <M> no es valida se borra la cinta, c.c se buscan los estados q_R y se reemplazan por un estado q, se agregan las transiciones para que loopee si para en q
Probar:

1) f es computable? *Rta:* Si, ya que M_f siempre se detiene al ser una entrada finita, luego de recorrerla se agregar/modifica un numero finito de tuplas y se detiene

2) (<M>,w) $\in L_u \Leftrightarrow$ (<M'>,w) $\in HP$

- **a)** (<M>,w) $\in L_u \Rightarrow$ (<M'>,w) $\in HP$
- **b)** (<M>,w) $\notin L_u \Rightarrow$ (<M'>,w) $\notin HP$

2a) (<M>,w) $\in L_u \Rightarrow$ (<M'>,w) $\in HP$
$\Rightarrow$ M acepta w
$\Rightarrow$ M para en q_A
$\Rightarrow$ M' para en q_A
$\Rightarrow$ M' se detiene con input w
$\Rightarrow$ (<M'>,w) $\in HP$

2b) (<M>,w) $\notin L_u \Rightarrow$ (<M'>,w) $\notin HP$

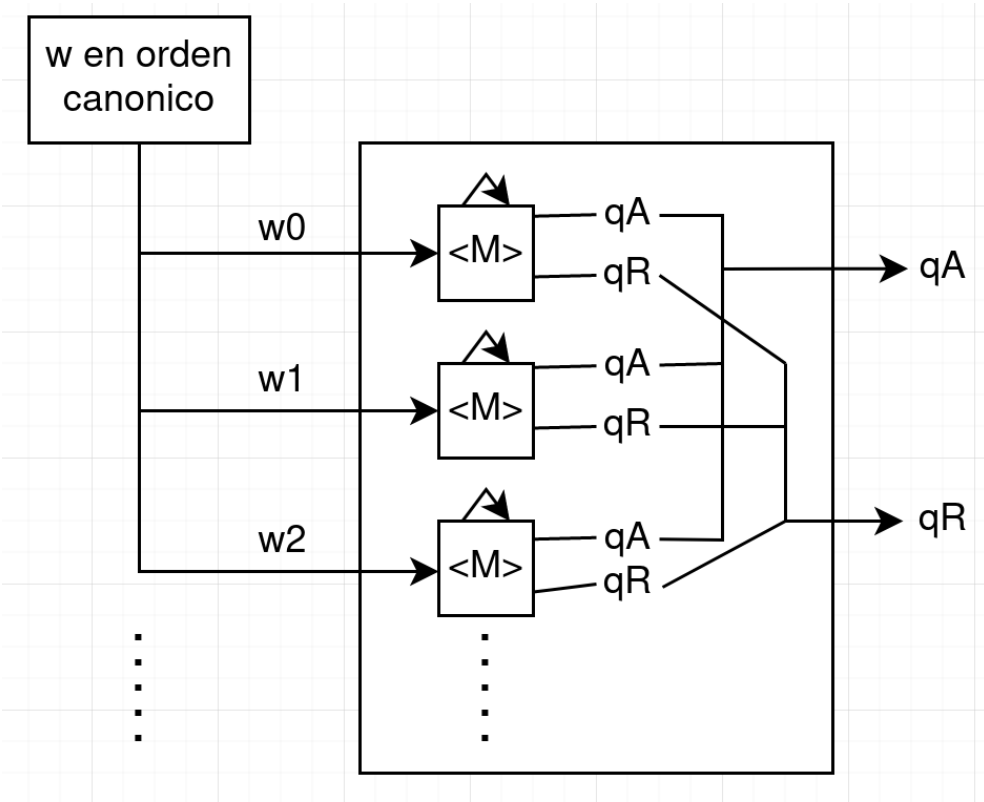
i. Si el par no es valido o <M> no es un codigo valido (<M'>,w) $= \lambda \Rightarrow$ (<M'>,w) $\notin HP$

ii. $\Rightarrow$ M no acepta w
$\Rightarrow$ M loopea o para en q_R
$\Rightarrow$ M' loopea en q
$\Rightarrow$ M' no se detiene con input w
$\Rightarrow$ (<M'>,w) $\notin HP$

(Ejercicio 13)

Consigna:
Demuestre que $L_{NV} \in$ RE
L_{NV} = {(<M>)/L(M)$\neq \varnothing$}
Respuesta:
Se puede demostrar creando una maquina M que simule con el codigo de maquina cada palabra del lenguaje en orden canónico, en un principio se simula una sola maquina j pasos con el primer w del lenguaje, una

vez se terminan los j pasos se mueve el cabezal y se simula con la maquina 2 con la segunda w haciendo j pasos en las dos maquinas, y asi sucesivamente hasta encontrar qA (entonces no es el lenguaje vacio), si el lenguaje es $\varnothing$ esta maquina va a rechazar cada w de forma infinita parando en qR. <M> tambien puede loopear con el w dado como input, por lo que queda demostrado que $L_{NV} \in RE$ y no acepta el lenguaje $\varnothing$.



Maquina de j pasos con MTs simulando cada palabra del lenguaje ordenado de forma canonica