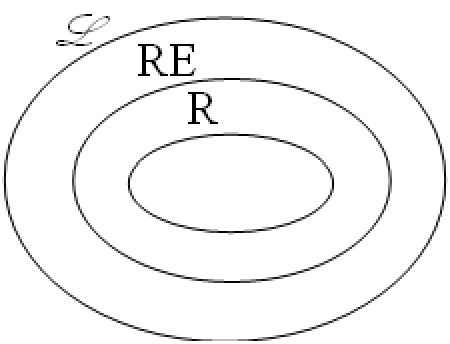
Caracterización de Lenguajes

Def.: un lenguaje L es recursivamente enumerable (RE) sii existe una MT M que lo acepte, es decir $L = L(M)$.

Def.: un lenguaje L es recursivo (R) o decidible sii existe una MT M tal que $L = L(M)$ y M siempre se detiene para todo input de Σ^*

Recordar: Dado un alfabeto Σ, denotamos con Σ^* al conjunto de todas las cadenas formadas por símbolos de Σ. $\mathscr{L}$ es el conjunto de todos los lenguajes definidos sobre el alfabeto Σ, es decir $\mathscr{L} = \rho(\Sigma^*)$, es decir el conjunto de todos los subconjuntos posibles de Σ^*. Se tiene la siguiente situación:



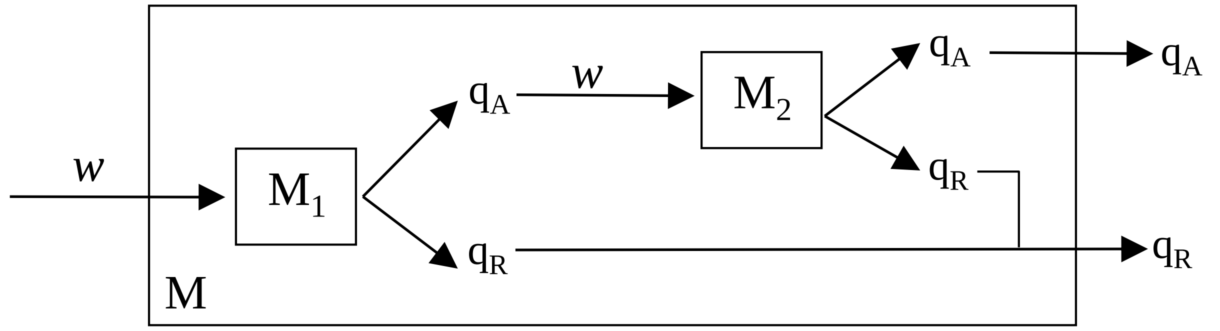
$$\underbrace{R \subseteq RE \subseteq \mathscr{L}}_{\text{por las definiciones}}$$

Caracterización de Lenguajes

Interrogantes: ¿Las inclusiones son propias?

$$\text{Es decir} \quad \begin{cases} ¿\mathscr{L} - \mathrm{RE} \neq \emptyset? \\ \\ ¿\mathrm{RE} - \mathrm{R} \neq \emptyset? \end{cases}$$

Ejercicio: Sean $L_1 \in R$ y $L_2 \in R$ ¿$L_1 \cap L_2 \in R$?



Rta.: sí $L_1 \cap L_2 \in R$

Dem.: Sean M_1 y M_2 MT de **una sola cinta** tq $L_1 = L(M_1)$ y $L_2 = L(M_2)$

Además se eligen ambas MT tal que se detienen para toda entrada, seguro existen porque ambos lenguajes pertenecen a R

$$M_1 = <Q_1, \Sigma, \Gamma_1, \delta^1, q_0^1, q_A^1, q_R^1>$$

$$\text{con } Q_1 \cap Q_2 = \varnothing$$

$$M_2 = <Q_2, \Sigma, \Gamma_2, \delta^2, q_0^2, q_A^2, q_R^2>$$

Se construye una MT de **dos cintas** que funciona de la siguiente manera:

1) Copia la entrada en la segunda cinta y posiciona el cabezal en el principio de la entrada en la 2da. cinta.

2) Simula M_1 sobre la cinta 2. Si M_1 para en q_R^1, M para en q_R, si M_1 para en q_A^1 ir al punto 3)

3) Borra la cinta 2

4) Copia w de la cinta 1 a la cinta 2.

5) Simula M_2 sobre w en la cinta 2.

Si M_2 para en q_R^2, M para en q_R

Si M_2 para en q_A^2, M para en q_A

¿ Cómo sería concretamente la codificación de esta MT ?

Para formalizar: $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R \rangle$

$\delta: Q \times \Gamma^2 \rightarrow Q \cup \{q_A, q_R\} \times (\Gamma \times \{D, I, S\})^2$

con $Q_1 \cup \{q^1_A, q^1_R\} \cup Q_2 \cup \{q^2_A, q^2_R\} \subseteq Q$; $\Gamma_1 \cup \Gamma_2 \subseteq \Gamma$

$\delta(q_i, a, b) = (q_j, (c, m_1), (d, m_2))$ con $q_i \in Q$; $q_j \in Q \cup \{q_A, q_R\}$; $a,b,c,d \in \Gamma$; $m_1, m_2 \in \{D, I, S\}$

Símbolos en cinta 1 y cinta 2 en la cinta 1 en la cinta 2

1) Copia la entrada en la segunda cinta y posiciona el cabezal en el principio de la entrada en la 2da. cinta.

$\delta(q_0, (x, B)) = (q_0, (x, D), (x, D))$ $(\forall x)(x \in \Sigma)$	(Copia la entrada en cinta 2)
$\delta(q_0, (B, B)) = (q_1, (B, S), (B, I))$	(Fin de copia, empieza a buscar el inicio del string en la cinta 2)
$\delta(q_1, (B, x)) = (q_1, (B, S), (x, I))$ $(\forall x)(x \in \Sigma)$	(se dirige al inicio del string en la cinta 2)
$\delta(q_1, (B, B)) = (q_0^1, (B, S), (B, D))$	(Queda apuntando al inicio del string en la cinta 2, para comenzar la simulación de M_1 sobre la cinta 2)

Para formalizar: $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R \rangle$

$\delta: Q \times \Gamma^2 \rightarrow Q \cup \{q_A, q_R\} \times (\Gamma \times \{D, I, S\})^2$

con $Q_1 \cup \{q^1{}_A, q^1{}_R\} \cup Q_2 \cup \{q^2{}_A, q^2{}_R\} \subseteq Q$; $\Gamma_1 \cup \Gamma_2 \subseteq \Gamma$

$\delta(q_i, \underbrace{a, b}_{\text{Símbolos en cinta 1 y cinta 2}}) = (q_j, (\underbrace{c, m_1}_{\text{en la cinta 1}}), (\underbrace{d, m_2}_{\text{en la cinta 2}}))$ con $q_i \in Q$; $q_j \in Q \cup \{q_A, q_R\}$; $a,b,c,d \in \Gamma$; $m_1, m_2 \in \{D, I, S\}$

2) Simula M_1 sobre la cinta 2. Si M_1 para en $q_R{}^1$, M para en q_R, si M_1 para en $q_A{}^1$ ir al punto 3)

Para cada $\delta^1(q_i{}^1, x) = (q_j{}^1, y, m)$ se define

$$\delta\,(q_i{}^1, (B, x)) = (q_j{}^1, (B, S), (y, m)) \qquad \text{(Simulación en cinta 2)}$$

$\delta(q_R{}^1, (B, x)) = (q_R, (B, S), (x, S)) \quad \forall x \in \Gamma_1$ (si M_1 Rechaza, M también)

$\delta(q_A{}^1, (B, x)) = (q_3, (B, S), (x, S)) \quad \forall x \in \Gamma_1$ (M_1 Acepta ir al punto 3)

q_3 es el estado en el que comienza la ejecución del punto 3)

Para formalizar: $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R \rangle$

$\delta: Q \times \Gamma^2 \rightarrow Q \cup \{q_A, q_R\} \times (\Gamma \times \{D, I, S\})^2$

con $Q_1 \cup \{q^1_A, q^1_R\} \cup Q_2 \cup \{q^2_A, q^2_R\} \subseteq Q$; $\Gamma_1 \cup \Gamma_2 \subseteq \Gamma$

$\delta(q_i, \underbrace{a, b}_{\text{Símbolos en cinta 1 y cinta 2}}) = (q_j, (\underbrace{c, m_1}_{\text{en la cinta 1}}), (\underbrace{d, m_2}_{\text{en la cinta 2}}))$ con $q_i \in Q$; $q_j \in Q \cup \{q_A, q_R\}$; $a,b,c,d \in \Gamma$; $m_1, m_2 \in \{D, I, S\}$

3) Borra la cinta 2

Pregunta: ¿Cómo saber cuánto borrar de la cinta 2? Tenga en cuenta que luego de simular M_1 la cinta posee cualquier string de $\Gamma_1{}^*$

Ejercicio1: De acuerdo a la estrategia elegida como respuesta a la pregunta anterior, completar la función delta de transición para los puntos 3), 4) y 5)

Ejercicio2: Demostrar que $L(M) = L(M_1) \cap L(M_2)$ y que M se detiene siempre. Con eso quedaría demostrado que $L_1 \cap L_2 \in R$.

Más definiciones

Def.: un lenguaje $L \in$ Co-R sii $\overline{L} \in R$ ($\overline{L}$ es el complemento de L respecto de Σ^*, es decir $\overline{L} = \Sigma^* - L$)

Def.: un lenguaje $L \in$ Co-RE sii $\overline{L} \in RE$

Más interrogantes:

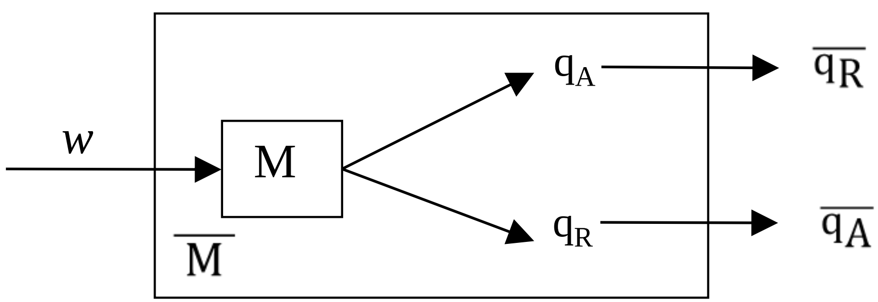
¿Qué relación habrá entre R, Co-R RE y Co-RE?

Teorema 1: R $\subseteq$ Co-R

Demostración. Hay que demostrar que si $L \in R \Rightarrow L \in$ Co-R, es decir que si $L \in R \Rightarrow \overline{L} \in R$.

Sea $L \in R \Rightarrow$ Existe una MT $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R \rangle$ tq $L = L(M)$ y M se detiene en algún momento para toda entrada.

Se construye $\overline{M} = \langle Q, \Sigma, \Gamma, \delta, q_0, \overline{q_A}, \overline{q_R} \rangle$ con $\overline{q_A} = q_R$ y $\overline{q_R} = q_A$



Hay que probar que $L(\overline{M}) = \overline{L}$ y además que $\overline{M}$ se detiene en algún momento para toda entrada.

1) Sea $w \in L(\overline{M}) \Leftrightarrow q_0w \vdash^{+}_{\bar{M}} \alpha_1\overline{q_A}\alpha_2 \Leftrightarrow q_0w \vdash^{+}_{M} \alpha_1 q_R\alpha_2 \Leftrightarrow w \notin L(M) \Leftrightarrow w \notin L \Leftrightarrow w \in \overline{L}$

def.L($\overline{M}$). Constr. def.L(M) Hip. def. $\overline{L}$

Por lo tanto, $L(\overline{M}) = \overline{L}$

2) ¿$\overline{M}$ se detiene para toda entrada?: Sí, por construcción $\overline{M}$ se detiene cuando M se detiene y por hipótesis M se detiene para toda entrada.

De 1) y 2) $L \in R \Rightarrow L \in$ Co-R

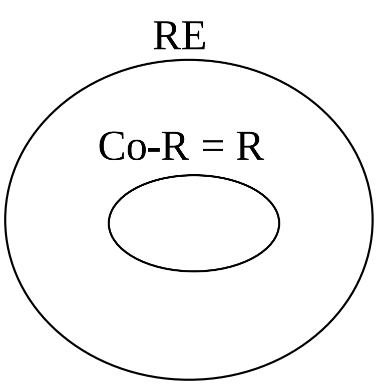
Por lo tanto, $R \subseteq$ Co-R

Teorema 2: Co-R $\subseteq$ R

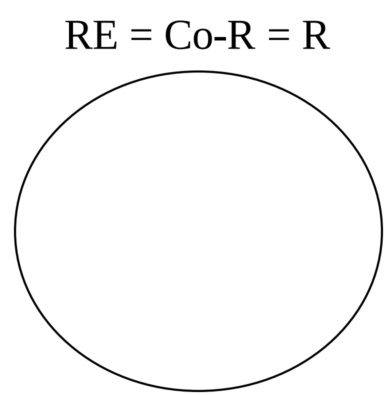
$$\text{Sea } L \in \text{Co-R} \underset{\text{def.Co-R}}{\Rightarrow} \overline{L} \in R \underset{\text{teor. ant.}}{\Rightarrow} \overline{L} \in \text{Co-R} \underset{\text{def. Co-R}}{\Rightarrow} \overline{\overline{L}} \in R \underset{\text{prop. de compl.}}{\Rightarrow} L \in R$$

Por lo tanto, Co-R $\subseteq$ R

Corolario: de los dos teoremas anteriores surge que R = Co-R



O bien



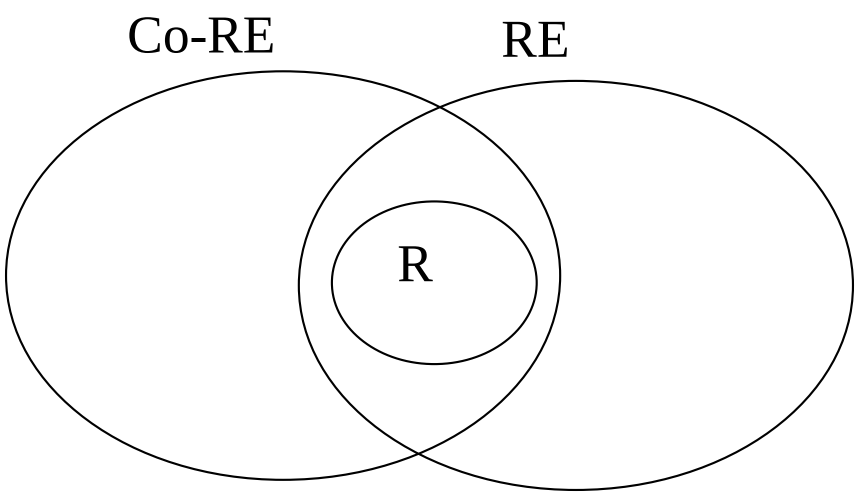
Teorema 3: R $\subseteq$ Co-RE

Dem.: Sea L $\in$ R $\Rightarrow$ $\overline{\text{L}}$ $\in$ R $\Rightarrow$ $\overline{\text{L}}$ $\in$ RE $\Rightarrow$ L $\in$ Co-RE

teor. 1 (def.R y RE) def. Co-RE

Por lo tanto, R $\subseteq$ Co-RE.

Además, como por definición R $\subseteq$ RE se tiene que R $\subseteq$ (RE $\cap$ Co-RE)



Teorema 4: $(RE \cap \text{Co-}RE) \subseteq R$

Sea $L \in (RE \cap \text{Co-}RE)$

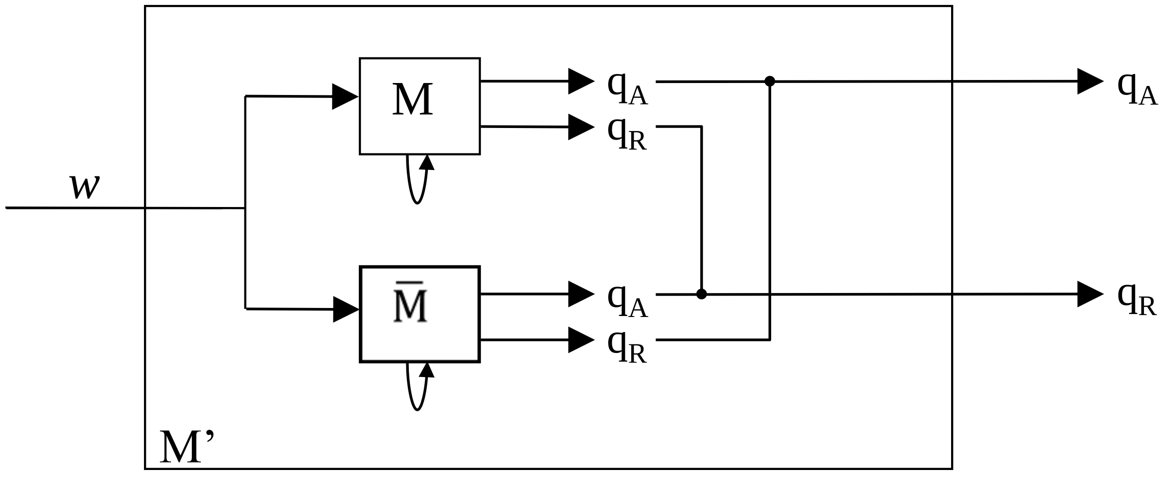
$\Rightarrow L \in RE \wedge \overline{L} \in RE$ (por def. de $\cap$ y de Co-RE)

$\Rightarrow$ existen M y $\overline{M}$, dos MT tq $L = L(M)$ y $\overline{L} = L(\overline{M})$

Hay que construir una MT M' que reconozca L y que se detenga siempre.

$\forall w \in \Sigma^*$ o bien $\overline{M}$ para en q_A o bien M para en q_A (no puede darse nunca el caso que ambas "loopeen").

Por lo tanto hay que construir M' simulando en paralelo M y $\overline{M}$, si M para en q_A o $\overline{M}$ para en $q_R \Rightarrow$ M' para en q_A y si M para en q_R o $\overline{M}$ en $q_A \Rightarrow$ M' para en q_R.



¿Cómo simular dos máquinas en paralelo? No importa la eficiencia, en una máquina de 4 cintas:

1) Escribir el número 1 en la cinta 4 (sea *i* ese valor).

2) Copiar w a las cintas 2 y 3.

3) Simular a lo sumo *i* pasos de M en la cinta 2 y a lo sumo *i* pasos de $\overline{M}$ en la cinta 3. si M para en q_A o $\overline{M}$ para en $q_R \Rightarrow$ M' para en q_A y si M para en q_R o $\overline{M}$ para en $q_A \Rightarrow$ M' para en q_R.

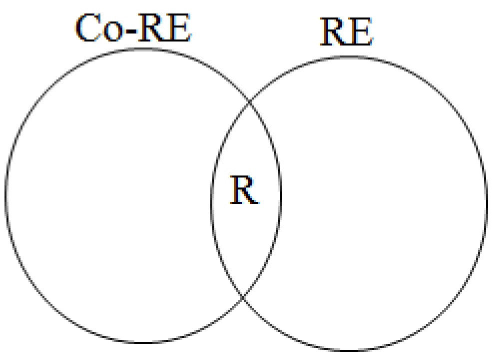
4) Borrar las cintas 2 y 3. Incrementar *i* en la cinta 4 y volver al punto 2.

Demostrar como ejercicio que L = L(M') y que M' se detiene siempre.

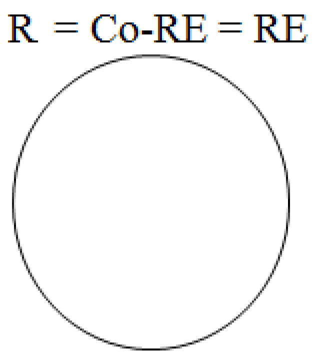
Pregunta: ¿Cómo se pueden simular *i* pasos?

Corolario: (RE $\cap$ Co-RE) = R (por los teoremas 3 y 4)

Por lo tanto hasta ahora nuestra situación es:



O bien



Def.: **orden canónico para Σ^*:** se listan todas las palabras en orden según su tamaño con las palabras del mismo tamaño en orden lexicográfico.

Ej: $\Sigma = \{0, 1\}$, el orden canónico es:

λ, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001 …

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

Obsérvese que si *w* es un string de $\{0,1\}^*$, la posición *i* que ocupa en el orden canónico se escribe en binario como 1*w*. Decimos entonces que *w* es el *i*-ésimo string y por ello lo denotamos w_i

Por ejemplo el string λ ocupa la posición 1 (1λ) el string 01 ocupa la posición 5 (101), el string 0000 la posición 16 (10000)

Pregunta 1: ¿Puede una MT generar las palabras de $\{0,1\}^*$ en orden canónico?

Rta.: Sí. Idea: ir sumando 1 en binario, cuando el resultado necesita un bit más, se ponen todos los bits en cero y se vuelve al proceso de sumar uno en binario.

Ejercicios para el lector

Ej. 1) construir una MT que escriba en la primera cinta las palabras de $\{0, 1\}^*$ en orden canónico separadas por ","

Ej. 2): construir una MT que genere las palabras de $\{a, b, c\}^*$ en orden canónico separadas por "," es decir: λ, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab ...

Pregunta 2: Si L es un lenguaje recursivo ($L \in$ R) ¿Puede una MT generar todas las palabras de L en orden canónico?

Rta.: Sí, porque existirá alguna MT M que reconoce L y siempre se detiene. Se construye una MT M' que va generando en una cinta los strings de Σ^* en orden canónico, simula M sobre cada string generado y si M lo acepta M' lo escribe en la cinta 1.

Pregunta 3: si L es un lenguaje recursivamente enumerable ($L \in$ RE) ¿Puede una MT generar todas las palabras de L?

Rta.: Sí. Se generan todos los pares (i, j) en orden de su suma, $i+j$, y entre los de igual suma en orden creciente de i (ver ejercicio de la práctica 1)

(1, 1); (1, 2); (2, 1); (1, 3); (2, 2); (3, 1), …

Por cada par (i, j) generado se simulan j pasos de la MT M que reconoce el lenguaje L (L = L(M)), sobre w_i (i-ésimo string de Σ^* en orden canónico). Si M acepta w_i en esos j pasos $\Rightarrow$ se escribe w_i en la cinta 1.

Pregunta 4: ¿Puede codificarse una MT como un string de un alfabeto de 2 símbolos?

Rta.: Sí. Se puede hacer de muchas formas, acá se muestra una.

Ej: se quiere codificar $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R \rangle$

$Q = \{q_0, q_1, \ldots, q_K\}$ $\Sigma = \{a_1, a_2, \ldots, a_L\}$ $\Gamma = \{B, a_1, a_2, \ldots, a_L, a_{L+1}, \ldots, a_n\}$

Se puede codificar M con un alfabeto binario {0, 1} de la sgte. forma:

Estados: $q_A = 1$ $q_R = 11$ $q_0 = 111$ $q_1 = 1111$ … $q_i = 1^{(i+3)}$

Símbolos: $B = 1$, $a_1 = 11$ $a_2 = 111$ … $a_i = 1^{(i+1)}$

Movimiento del Cabezal: D = 1 I = 11 S = 111

Función de Transición: cada transición se codifica como una quíntupla de elementos separados por un símbolo 0. Ej.:

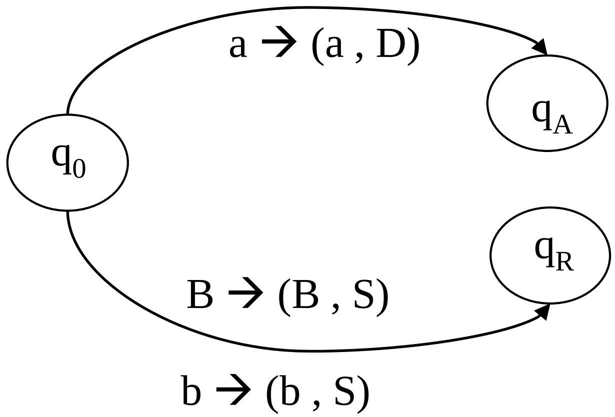
$\delta(q_0, a_2) = (q_1, a_4, D)$ se codifica como

$(q_0, a_2, q_1, a_4, D) = (11101110111101111101)$

q_0, a_2, q_1, a_4, D

M se codifica como una sucesión de quíntuplas separadas por 00.

Ej.: M = $<Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R>$ $Q = \{q_0\}$ $\Sigma = \{a, b\}$ $\Gamma = \{B, a, b\}$



$d(q_0, a) = (q_A, a, D)$
$d(q_0, b) = (q_R, b, S)$
$d(q_0, B) = (q_R, B, S)$

Ej.: M = $<Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R>$ $Q = \{q_0\}$ $\Sigma = \{a, b\}$ $\Gamma = \{B, a, b\}$

$q_A = 1$	$q_R = 11$	$q_0 = 111$
$B = 1$	$a = 11$	$b = 111$
$D = 1$	$I = 11$	$S = 111$

$<M>$ = 1110110101101001110111011011101110011101011010111

$\delta(q_0, a) = (q_A, a, D)$ $\delta(q_0, b) = (q_R, b, S)$ $\delta(q_0, B) = (q_R, b, S)$

Note que existen otros posibles códigos para M. Las tres transiciones pueden listarse en distinto orden (3! formas distintas) por lo tanto hay 6 códigos distintos para la misma M

Para pensar

¿Cuál sería la codificación de MT que tiene menor valor numérico?

Es decir: toda codificación de MT puede verse como un número binario, ¿cuál sería la MT codificada de "menor" valor numérico?

Todas tienen:

- Los estados q_0, q_A y q_R
- Al menos 2 símbolos en Γ: B y 1 símbolo de Σ
- δ debe ser completa δ: $Q \times \Gamma \rightarrow Q \cup \{q_A, q_R\} \times \Gamma \times \{D, I\}$, completa

$M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R \rangle$ $\quad Q = \{q_0\}$ $\quad \Sigma = \{a\}$ $\quad \Gamma = \{B, a\}$

$q_A = 1$	$q_R = 11$	$q_0 = 111$
$B = 1$	$a = 11$	
$D = 1$	$I = 11$	$S = 111$

$\langle M \rangle = 1110110101101001110101010111$

$\delta(q_0, a) = (q_A, a, D)$ $\quad \delta(q_0, B) = (q_A, B, S)$

¿Esta es la codificación de menor valor numérico?

Pregunta 5: ¿$L_{<M>} \in$ R?
Con $L_{<M>} = \{w \in \{0, 1\}^* \,/\, w$ es el código bien formado de una MT$\}$

Rta.: Sí, para demostrarlo hay que construir una MT que realice un chequeo sintáctico y siempre se detenga.

El código binario de una MT M se denotará con <M>

Def. Se denomina *i*-ésima MT y se denota M_i a la MT cuyo código es w_i (*i*-ésimo string binario en orden canónico), es decir $<M_i> = w_i$

Convención: Si w_i no es un código válido de MT se considera que codifica a M_i, siendo M_i una MT que se detiene inmediatamente rechazando cualquier entrada, $L(M_i)=\{\}$. Así, todo string w_i de $\{0,1\}^*$ se corresponde con, o codifica, una MT M_i

Def.: se define el lenguaje Diagonal L_D (primer ejemplo de lenguaje no recursivamente enumerable que se verá) de la siguiente manera:

$$L_D = \{w_i \in \Sigma^* \,/\, w_i \notin L(M_i)\}$$

siendo $\Sigma = \{0,1\}$; w_i el *i*-ésimo string en orden canónico de Σ^* y M_i la *i*-ésima MT.

L_D: codificaciones de MT que rechazan su propia codificación

$L_D = \{w_i \in \Sigma^* \ / \ w_i \notin \mathrm{L}(\mathrm{M}_i)\}$

Teorema: $L_D \notin$ RE

Dem.: Por reducción al absurdo, se asume que $L_D \in$ RE $\Rightarrow$ existirá una MT M que lo acepta y cuya codificación está en alguna posición del orden canónico, es decir que existe algún natural k para el cual M = M_k y $L_D = \mathrm{L}(\mathrm{M}_k)$.

Considerando w_k el k-ésimo string de Σ^*, hay dos posibilidades, o bien $w_k \in L_D$ o bien $w_k \notin L_D$

a) $w_k \in L_D \Rightarrow w_k \notin \mathrm{L}(\mathrm{M}_k) \Rightarrow$ (por def. L_D)

$\Rightarrow w_k \notin L_D$ (porque $\mathrm{L}(\mathrm{M}_k) = L_D$)

(contradicción)

b) $w_k \notin L_D \Rightarrow w_k \in \mathrm{L}(\mathrm{M}_k)$ (por def. L_D)

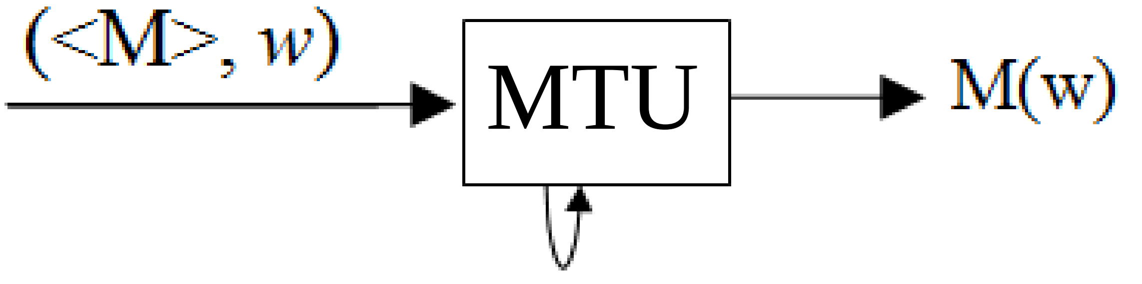
$\Rightarrow w_k \in L_D$ (porque $\mathrm{L}(\mathrm{M}_k) = L_D$)

(contradicción)

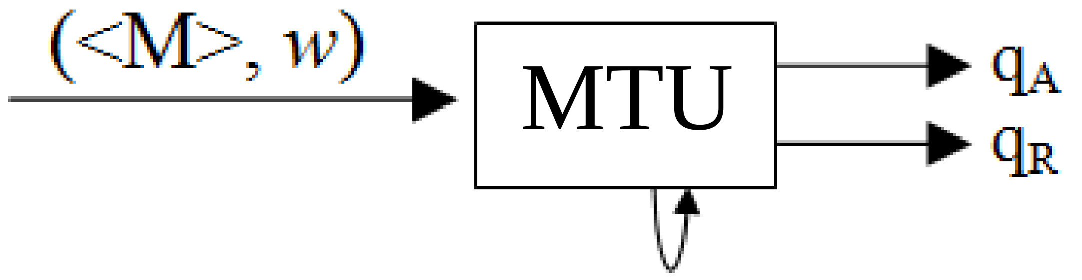
En ambos casos se llega a una contradicción y por ende no puede existir una M_k que reconozca al lenguaje L_D, por lo tanto L_D no puede ser recursivamente enumerable.

Por lo tanto $L_D \notin$ RE.

Máquina de Turing Universal (MTU): es una MT que recibe como entrada la codificación de otra MT *M* y una entrada *w*. La MTU ejecuta la MT M sobre la entrada *w*.



Nota: también se tiene una versión reconocedora de lenguajes



Esta máquina responde a cuestiones relativas a otras MT

Claramente la MTU puede construirse puesto que el proceso de mirar el estado y el símbolo corriente, buscar la quíntupla de δ que se va a aplicar y realizar lo que indica o detenerse es un procedimiento efectivo.

Se puede asumir que la entrada *w* se separa del código de la MT con 000, y M se codifica de la manera que ya se ha visto.

Una idea de cómo construir una MTU

Se copia *w* en la cinta 2 sobre la que se realiza la simulación

Es necesario identificar:

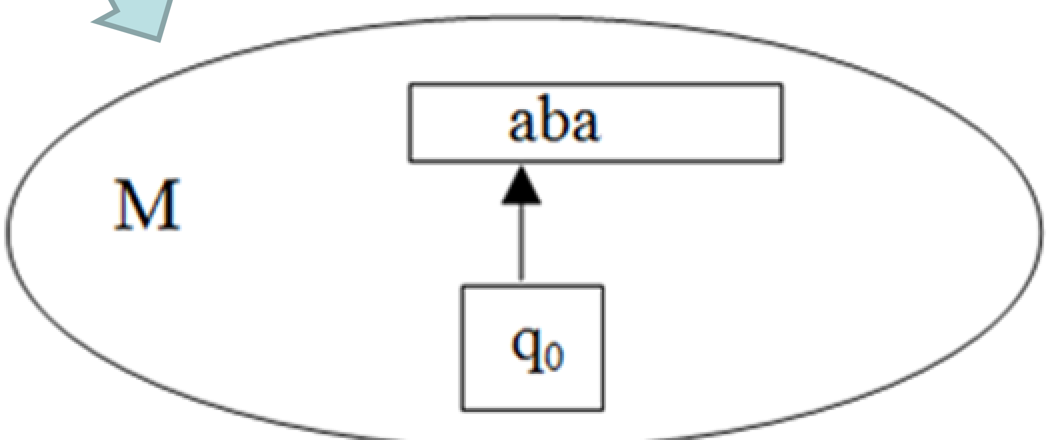
- Posición del cabezal (se puede usar una tercera cinta)
- Estado actual (se puede usar una cuarta cinta)
- Símbolo actualmente leído (se puede usar una quinta cinta)

MUT de 5 cintas

$\overbrace{\texttt{1110110101010}0\ldots}^{<M>}\ \texttt{000}\ \overbrace{\texttt{110111011}}^{w}$

11101101010100 ... 000110111011	Entrada
110111011	Cinta de Simulación
1	Posición del Cabezal
111	Estado Actual
11	Símbolo Actual

Representa esta situación

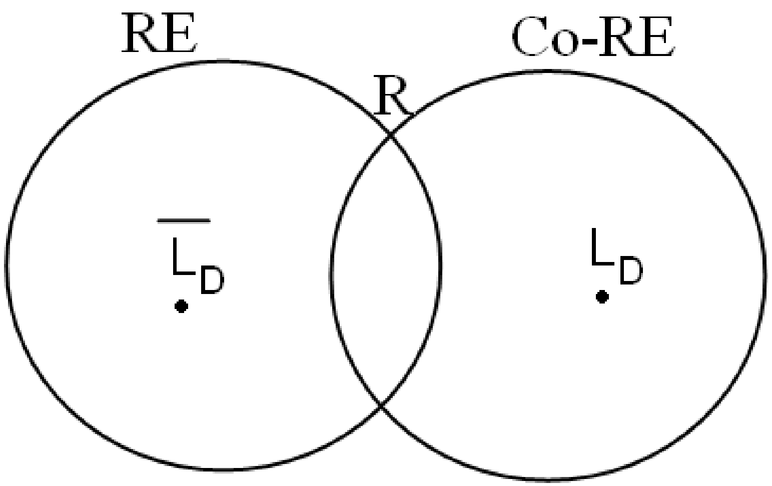


Nota1: puede probarse que L_D pertenece a Co-RE, pues fácilmente se verifica que:

$$\overline{L_D} \in \text{RE} \quad \left(\overline{L_D} = \{w_i \in \Sigma^* \,/\, w_i \in \text{L}(\text{M}_i)\} \right)$$

Para ello se construye una MT que utilizando una MTU acepta $\overline{L_D}$ ejecutando el código de M_i sobre w_i aceptando sii M_i acepta w_i. Recuérdese que el código $<\text{M}_i>$ de la MT M_i se obtiene fácilmente pues $<\text{M}_i> = w_i$

Nota 2: Además $\overline{L_D}$ no puede estar en R, ya que esto implicaría que L_D también esté en R (y ya sabemos que L_D no pertenece a RE). Por lo tanto $\overline{L_D} \in$ RE - R.



Def.: se define L_u , el lenguaje universal, como:

$$L_u = \{(\langle M\rangle, w) \;/\; M \text{ acepta } w\}$$

Pregunta: ¿$L_u \in$ RE?

Rta.: claramente sí. Se puede construir M_u de la siguiente manera:

1) Si ($\langle M\rangle$, w) no es un par válido parar en q_R

2) En caso contrario separar $\langle M\rangle$ de w

3) Si $\langle M\rangle$ es un código inválido parar en q_R

4) Simular M sobre w. Si M para en $q_A \Rightarrow M_u$ para en q_A. Si M para en $q_R \Rightarrow M_u$ para en q_R. Si M loopea $\Rightarrow M_u$ también loopea.

Claramente $L_u = L(M_u)$, por lo tanto $L_u \in$ RE

¿$L_u \in$ R?

Se verá que no es cierto probando que $\overline{L_u} \notin$ RE

L_u = {(<M>, w) / M acepta w} entonces :

(<M>, w) $\in \overline{L_u}$ sii (<M>, w) no es un par válido o M rechaza w

Teorema: $\overline{L_u} \notin$ RE

Dem.: se demuestra que si $\overline{L_u} \in$ RE $\Rightarrow L_D \in$ RE, lo cual es absurdo pues ya se vio que $L_D \notin$ RE.

Si $\overline{L_u} \in$ RE $\Rightarrow \exists$ MT $\overline{\mathrm{M}_u}$ que acepta $\overline{L_u}$, es decir que dada una entrada (<M>, w) puede identificar que M rechaza w

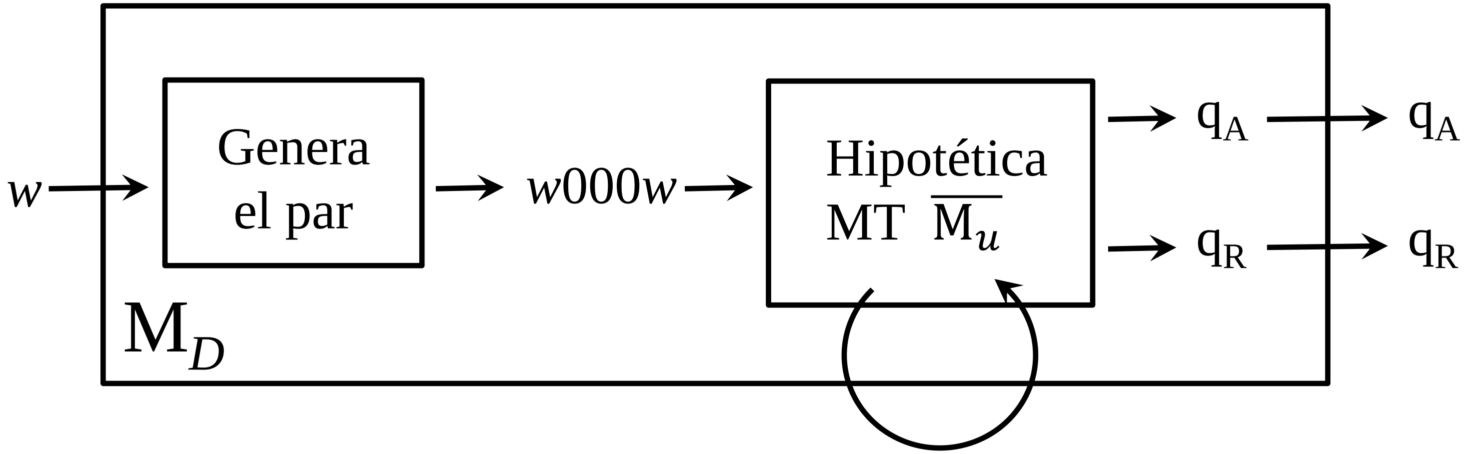
Considerando que existe tal $\overline{\mathrm{M}_u}$ se construye la MT M_D que acepta L_D de la siguiente manera:

1) M_D a partir de la entrada w genera el par $w000w$ (recordar que $<\mathrm{M}_i> = w_i$)

2) M_D simula $\overline{\mathrm{M}_u}$ sobre $w000w$. M_D acepta w sii $\overline{\mathrm{M}_u}$ acepta $w000w$

$L_u = \{(<M>, w) \,/\, M \text{ acepta } w\}$ entonces :

$(<M>, w) \in \overline{L_u}$ sii $(<M>, w)$ no es un par válido o <u>M rechaza w</u>



Si w es w_i en nuestra numeración se tiene que:

M_D acepta $w_i \Leftrightarrow \overline{M_u}$ acepta $<M_i>000w_i \Leftrightarrow M_i$ rechaza $w_i \Leftrightarrow w_i \in L_D$

De esta forma $L(M_D) = L_D$ lo que significa que $L_D \in$ RE

Por lo tanto demostramos que si $\overline{L_u} \in \text{RE} \Rightarrow L_D \in \text{RE}$,

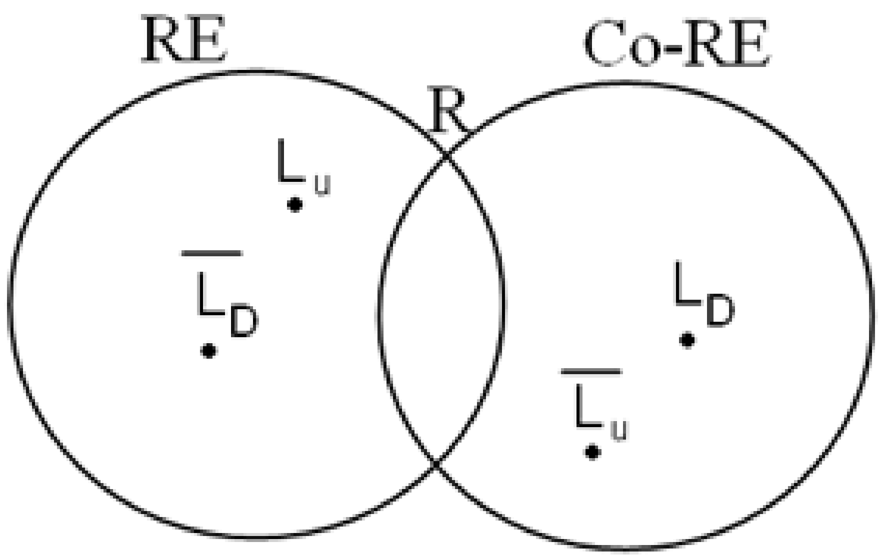
Por contrarrecíproca se tiene que $L_D \notin \text{RE} \Rightarrow \overline{L_u} \notin \text{RE}$

y como ya se conoce que $L_D \notin$ RE se tiene que $\boldsymbol{\overline{L_u} \notin \textbf{RE}}$

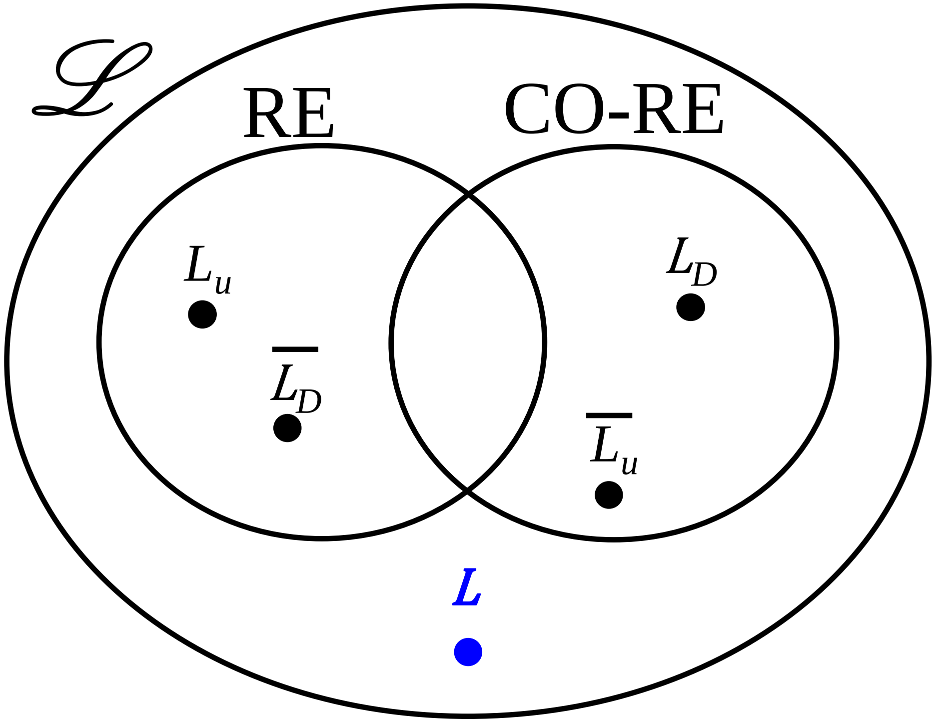
Corolario: $L_u \in$ (RE – R).

Inmediato pues ya sabemos que L_u está en RE y que L_u no puede estar en R pues ello implicaría que $\overline{L_u}$ también estuviese en R lo que sería un absurdo pues acabamos de demostrar que $\overline{L_u} \notin$ RE

Hasta acá se tiene entonces la siguiente situación:



Lenguaje del conjunto $\mathscr{L} - (\text{RE} \cup \text{CO-RE})$



$$L = \{1w \,/\, w \in L_D\} \cup \{0w \,/\, w \notin L_D\}$$

$$L \in \mathscr{L} - (\text{RE} \cup \text{CO-RE})$$

Demostración

Claramente $L \in \mathscr{L}$. Falta mostrar que $L \notin$ (RE $\cup$ CO-RE), es decir que $L \notin$ RE y que $L \notin$ Co-RE

a) Demostrar que $L \notin$ RE

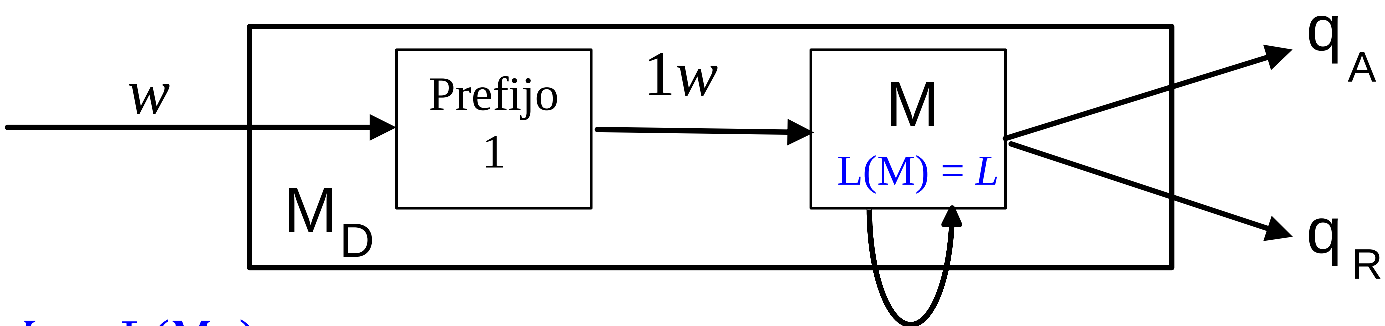
Vamos a demostrar que si $L \in$ RE entonces $L_D \in$ RE (absurdo, ya demostramos que $L_D \notin$ RE) por lo tanto la hipótesis $L \in$ RE no puede ser cierta.

Si $L \in$ RE entonces existe una MT M que acepta L. Vamos a construir una MT M_D que acepte L_D trabajando de esta manera:

1) A partir de una entrada w genera la cadena $1w$ en la cinta

2) Simula M sobre $1w$, respondiendo como M con entrada $1w$

$L = \{1w \mid w \in L_D\} \cup \{0w \mid w \notin L_D\}$



Veamos que $L_D = L(M_D)$

(1) $w \in L_D \Rightarrow 1w \in L$ (por definición de L)

$\Rightarrow$ M acepta $1w$ (porque M reconoce L)

$\Rightarrow M_D$ acepta w (por construcción de M_D)

$\Rightarrow w \in L(M_D)$

(2) $w \notin L_D \Rightarrow 1w \notin L$ (por definición de L)

$\Rightarrow$ M rechaza $1w$ (porque M reconoce L)

$\Rightarrow M_D$ rechaza w (por construcción de M_D)

$\Rightarrow w \notin L(M_D)$

por el contra-recíproca $w \in L(M_D) \Rightarrow w \in L_D$

De (1) y (2) $w \in L_D \Leftrightarrow w \in L(M_D)$ **Por lo tanto $L_D = L(M_D)$**

Así se demostró que si $L \in RE \Rightarrow L_D \in RE$

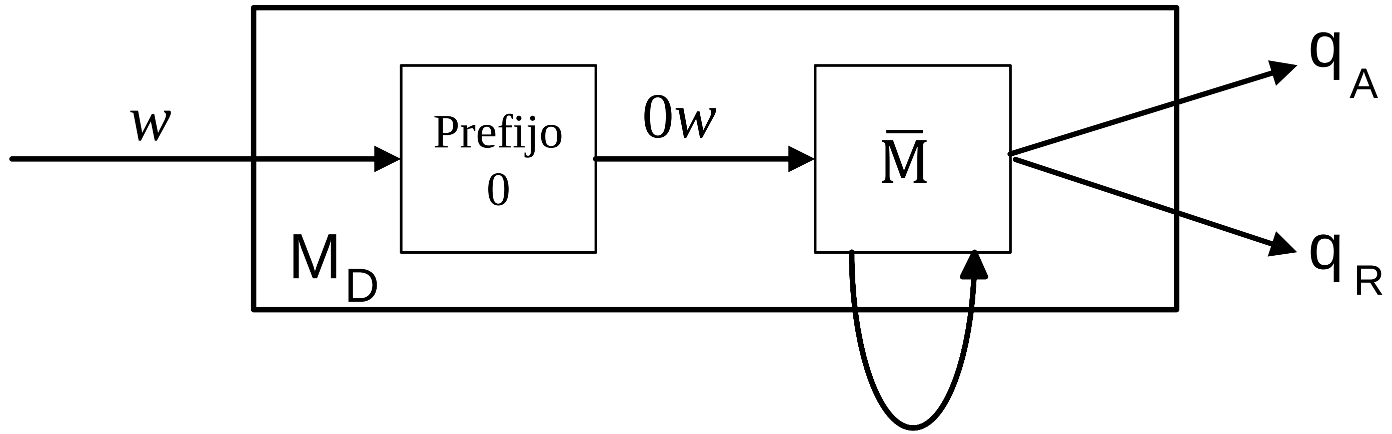
Pero ya sabemos que $L_D \notin RE$ por lo tanto $\boldsymbol{L \notin RE}$

b) Demostrar que $L \notin$ CO-RE

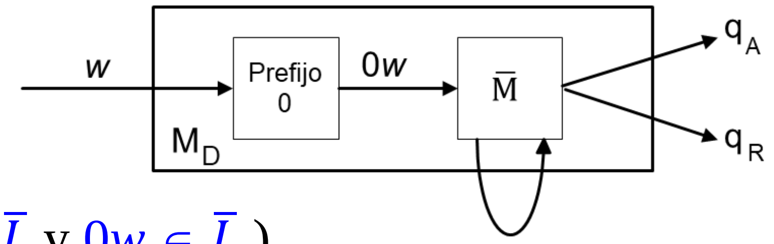
Vamos a demostrar que si $L \in$ CO-RE entonces $L_D \in$ RE (absurdo, ya demostramos que $L_D \notin$ RE) por lo tanto la hipótesis $L \in$ CO-RE no puede ser cierta.

Si $L \in$ CO-RE entonces $\overline{L} \in$ RE, entonces existe una MT $\overline{M}$ que acepta $\overline{L}$. Vamos a construir una MT M_D que acepte L_D trabajando de esta manera:

1) A partir de una entrada w genera la cadena $0w$ en la cinta
2) Simula $\overline{M}$ sobre $0w$, respondiendo como $\overline{M}$ con entrada $0w$



$L = \{1w \,/\, w \in L_D\} \cup \{0w \,/\, w \notin L_D\}$



Veamos que $L_D = L(M_D)$

(1) $w \in L_D \Rightarrow 0w \notin L$ (por definición de L)

$\Rightarrow \overline{M}$ acepta $0w$ (porque $\overline{M}$ reconoce $\overline{L}$ y $0w \in \overline{L}$)

$\Rightarrow M_D$ acepta w (por construcción de M_D)

$\Rightarrow w \in L(M_D)$

(2) $w \notin L_D \Rightarrow 0w \in L$ (por definición de L)

$\Rightarrow \overline{M}$ rechaza $0w$ (porque $\overline{M}$ reconoce $\overline{L}$ y $0w \notin \overline{L}$)

$\Rightarrow M_D$ rechaza w (por construcción de M_D)

$\Rightarrow w \notin L(M_D)$

por el contra-recíproca $w \in L(M_D) \Rightarrow w \in L_D$

De (1) y (2) $w \in L_D \Leftrightarrow w \in L(M_D)$ **Por lo tanto $L_D = L(M_D)$**

Así se demostró que si $L \in$ CO-RE $\Rightarrow L_D \in$ RE

Pero ya sabemos que $L_D \notin$ RE por lo tanto **$L \notin$ CO-RE**

Por lo demostrado en **a)** y **b)** se tiene que

$L \notin (\text{RE} \cup \text{CO-RE})$