Practica 3

1) Construir máquinas de Turing que acepten los siguientes lenguajes
   a) L1 = Σ*

      δ = { (q0, n) → (qA, n, S) / n ∈ Γ }

   b) L2 = {λ}

      δ = { (q0, B) → (qA, B, S),
      (q0, n) → (qR, n, S) / n ∈ Σ }

   c) L3 = ∅

      δ = { (q0, n) → (qR, n, S) / n ∈ Γ }

   d) L4 = $\{0^n 1^{2n}$ / n ≥ 0 }

      q0, 0
      q1, B, D

      q1, 0
      q1, 0, D

      q1, 1
      q1, 1, D

      q1, B
      q2, B, I

      q2, 1
      q3, B, I

      q2, 0
      qR, 0, S

      q2, B
      qR, B, S

q3, 1
q4, B, I

q3, 0
qR, 0, S

q3, B
qR, B, S

q4, 1
q4, 1, I

q4, 0
q4, 0, I

q4, B
q0, B, D

q0, B
qA, B, S

q0, 1
qR, 1, S

e) L5= {a^n b^n c^n / n ≥ 0 }

q0, a, B
q1, a, D, a,D

q1, a, B
q1, a,D, a,D

q1, b, B
q2, b,S, B,I

q2, b, a
q2, b,D, a,I

q2, b, B
qR, b,S,B,S

q2, c,B
q3,c,S,B,S

q2,c,a
qR,c,S,a,S

q3, c, B
q4, c,S,B,D

q4, c, a
q4, c,D,a,D

q4, B, a
qR, B,S,a,S

q4, B, B
qA, B,S, B,S

q4, c, B
qR, c,S,B,D

f) L6= { $a^n b^m c^k$ / k = n + m, n, m ≥ 1 }

q0, B, B
qR, B,S, B,S

q0, a, B
q1, a,D, a,D

q0, b, B
qR, b,S, B,S

q1, b, B

q1, b,D, b,D

q1, a, B
q1, a,D, b,D

q1, B, B
qR, B,S, B,S

q1, c, B
q2, c,S, B,I

q2, c, b
q2, c,D, b,I

q2, c, a
q2, c,D, a,I

q2, B, B
qA, B,S, B,S

q2, c, B
qR, c,S, B,S

q2, B, a
qR, B,S, a,S

q2, B, b
qR, B,S, b,S

g) L7 = {ww^R / w ∈ {0,1}*}, dónde w^R es el reverso de w

q0, 0
q1, B, D

q0, B
qR, B, D

q0, 1

q2, B, D

q1, 0

q1, 0 , D

q1, 1

q1, 1 , D

q2, 0

q2, 0 , D

q2, 1

q2, 1 , D

q1, B

q3, B, I

q2, B

q4, B, I

q4, 0

qR, 0,S

q4, 1

q10, B,I

q4, B

qR, B, S

q10, 1

q10, 1, I

q10, 0

q10,0, I

q10, B

q11, B, D

q3, 1
qR, 1,S

q3, 0
q10, B,I

q3, B
qR, B, S

q10, 1
q10, 1, I

q10, 0
q10,0, I

q10, B
q11, B, D

q11, 0
q1, B, D

q11, 1
q2, B, D

q11, B
qA, B, S

h) L8= L7 ∪ $\{w^0w^R$ / w ∈ {0,1}*} ∪ $\{w^1w^R$ / w ∈ {0,1}*}

q0, 1, B
q0, 1,D, 1,D

q0, 0, B
q0, 0,D, 0,D

q0, B, B
q1, B,I, B,I

q1, 0, 0
q1, 0,S, 0,I

q1, 1, 1
q1, 1,S, 1,I

q1, 0, 1
q1, 1,S, 1,I

q1, 1, 0
q1, 1,S, 0,I

q1, 1, B
q2, 1,S, B,D

q1, 0, B
q2, 0,S, B,D

q2, 1, 1
q2, 1,I,1,D

q2, 0, 0
q2, 0,I,0,D

q2, 1, 0
qR, 1,S,0,S

q2, 0, 1
qR, 0,S,1,S

q2, B, B
qA, B,S,B,S

2) Construya una Máquina de Turing de 2 cintas que implemente un contador binario en la segunda cinta para contabilizar la cantidad de letras "a" que aparecen en el input de la primera cinta. Con Σ={ a, b}; Γ={ a, b, 0, 1, B}

q0, a, B
q1, a,S,B,S

q0, a, 0
q1, a,S,0,S

q0, a, 1
q1, a,S,1,S

q0, b, B
q0, b,D,B,S

q0, b, 1
q0, b,D,1,S

q0, b, 0
q0, b,D,0,S

q0, B,B
qd, B,S,B,S

q0, B,0
qd, B,S,0,S

q0, B,1
qd, B,S,1,S

q1, a, B
q0, a,D,1,S

q1, a, 0
q0, a,D,1,S

q1, a, 1
q2, a,S,0,I

q2, a, 1
q2, a,S,0,I

q2, a, 0
q3, a,S,1,D

q2, a, B
q3, a,S,1,D

q3, a, 0
q3, a,S,0,D

q3, a, B
q0, a,D,B,I

3) Sea M una máquina de Turing del modelo "D-I-S" (Derecha, Izquierda, Estático). ¿Existe una máquina de Turing M' equivalente a M que comience con el cabezal apuntando a cualquier celda de la cinta? Note que M' puede apuntar a una celda no ocupada por el input. ¿Qué puede decir al respecto si se sabe que $\lambda \notin L(M)$? Justifique su respuesta.

Si λ pertenece a L(M), entonces cuando M' comienza su ejecución en cualquier celda de la cinta, no puede determinar de inmediato si la cadena de entrada es vacía o no. Esto puede llevar a que M' quede en un bucle infinito si se tratara de que la entrada fuera λ, lo que implicaría que M' estaría rechazando a λ mientras que M la estaría aceptando, lo que hace que M y M' no sean equivalentes en este caso.

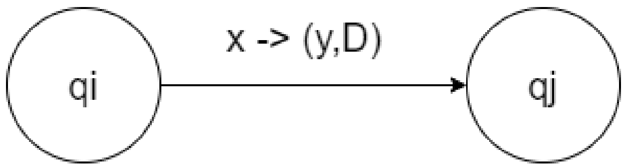
Por otro lado, si λ no pertenece a L(M), sí es posible crear una máquina de Turing M' equivalente a M. En este caso, podemos encontrar la cadena original (si es que no caemos dentro) haciendo un movimiento de derecha a izquierda a través de la cinta escribiendo una cadena especial que sirva como marcador para determinar las celdas por las que ya hemos pasado y las que no. Luego, borramos la cadena generada durante el proceso de recorrido y posicionamos el cabezal en la posición inicial de la cadena para que comience la ejecución. A partir de esta instancia, M' utiliza las transiciones originales de M para procesar la entrada y simular su comportamiento.

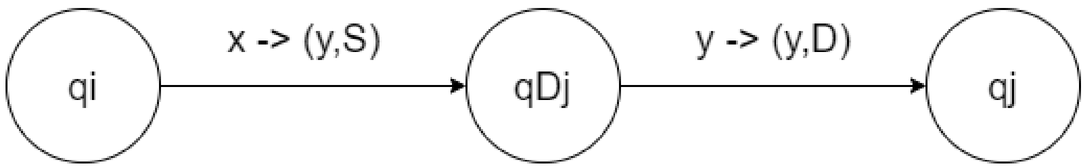
4) Probar que para toda máquina de Turing M reconocedora del modelo DIS, existe una máquina de Turing M' equivalente con la restricción que no puede cambiar el símbolo de la cinta y mover el cabezal simultáneamente, es decir: $\delta'(q_i , a_k) = (q_j , a_l , X)$, si $a_k \neq a_l$ entonces X=S

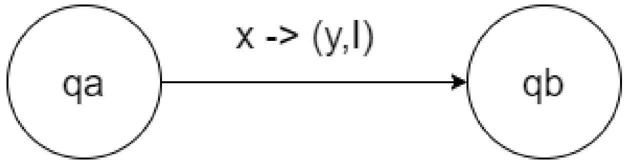
M = <Q, Σ, Γ, δ, q0, qA, qR>
M’ = <Q’, Σ, Γ, δ’, q0, qA, qR>
δ': Q’x Γ → Q’ U {qA, qR} x Γ x {D, I,S}

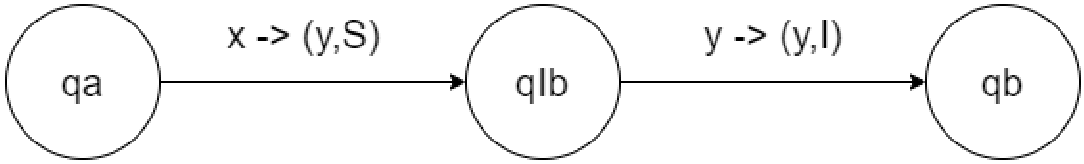
a) Ambos empiezan en el estado q0. x, y ∈ Γ. x =/ y. qDj, qIb ∈ Q’. (estados que recuerdan cosas)
b) Se agregan las siguientes transiciones (solo reemplazan a las transiciones de M que rompen con la regla de M’)

Cuando se quiere cambiar el símbolo y mover el cabezal:









Básicamente hay un estado intermedio que depende de la dirección a donde me quiero mover y del estado al cual voy a pasar (para poder recordar a donde debo ir). Como el símbolo lo cambio antes de pasar a ese estado intermedio, no hace falta recordarlo. Va a haber tantos estados intermedios como direcciones y estados debo recordar.

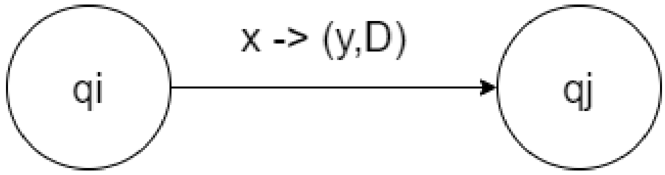
La demostración que L(M) = L(M)’ es muy similar a lo visto en la teoría.

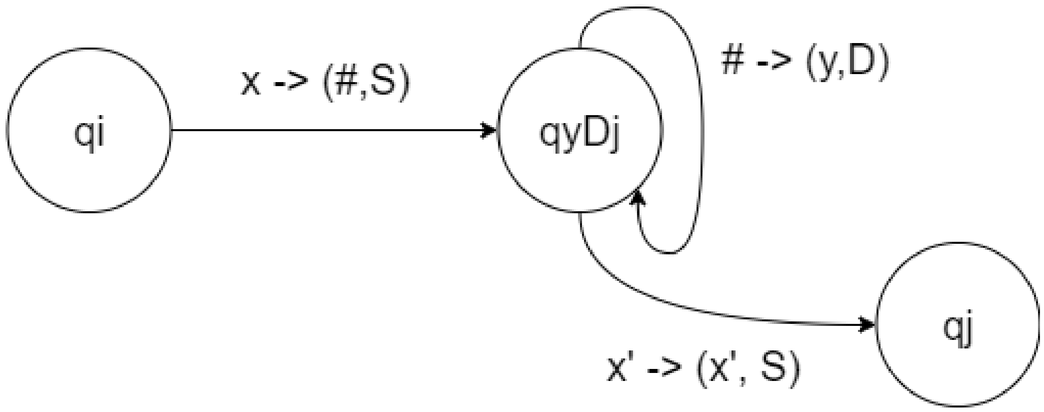
5) Probar que para toda máquina de Turing M reconocedora del modelo DIS, existe una máquina de Turing M' equivalente con la restricción que no puede cambiar de estado y mover el cabezal simultáneamente, es decir:
$\delta'(q_i , a_k) = (q_j , a_l , X)$, si $X \in\{D,I\}$ entonces $i=j$

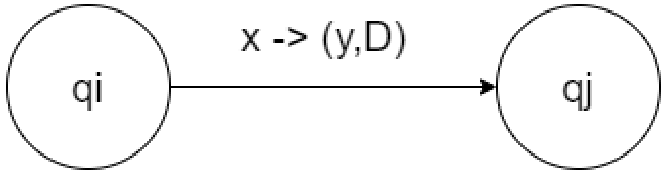
M = <Q, Σ, Γ, δ, q0, qA, qR>
M’ = <Q’, Σ, Γ’, δ’, q0, qA, qR>
δ': Q’x Γ → Q’ U {qA, qR} x Γ x {D, I,S}

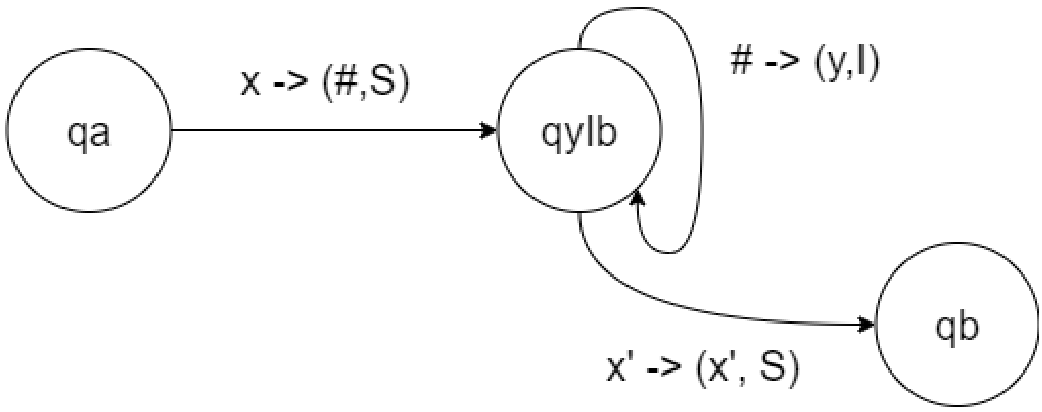
a) Ambos empiezan en q0. x, y, x’ ∈ Γ. x no necesariamente es distinto a y. qyDj, qyIb ∈ Q’. # ∈ Γ‘.
b) Se agregan las siguientes transiciones (solo reemplazan a las transiciones de M que rompen con la regla de M’)

Si se quiere cambiar de estado y mover el cabezal.









Estado intermedio que recuerda a que dirección me debo mover (y que debo escribir ya que estoy reemplazándolo por el símbolo #) y a que estado debo pasar.

La demostración que L(M) = L(M)’ es muy similar a lo visto en la teoría.