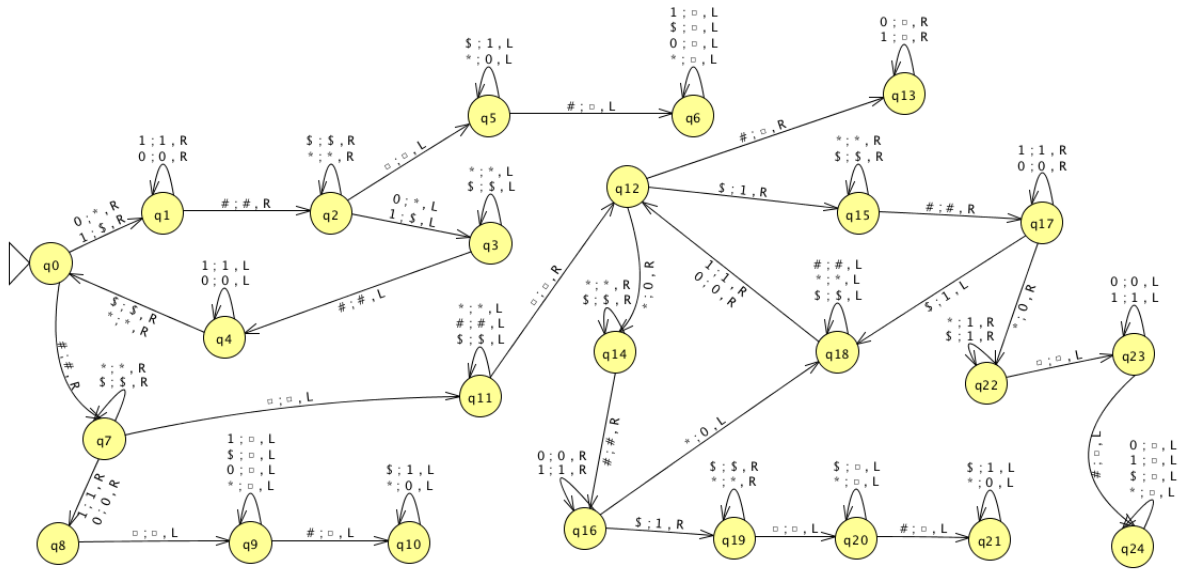


Tarea 4 - TALF

Jorge Contreras 201573547-6
Juan Pablo Jorquera 201573533-6

1. Pregunta 1

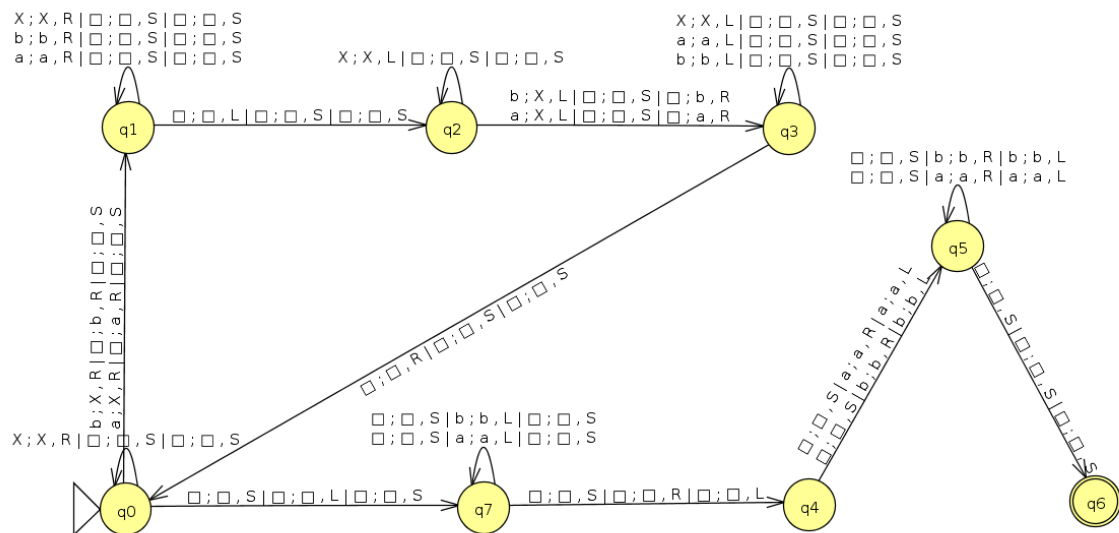


La máquina de Turing se divide en dos grandes partes: inicialmente compara los tamaños para decidir cual es el menor y luego, de ser iguales compara término a término para definir el menor.

- Entre q_0 y q_4 va marcando los 1 con \$ y los 0 con * y luego verifica que exista ese dígito en el término de la derecha.
- En q_2 se sale al haber encontrado un dígito a la izquierda y haber terminado por completo el de la derecha, entonces el de la derecha es más corto, y en q_5 y q_6 lo decodifica y limpia el resto.
- En q_0 se sale al haber terminado el primer término, entonces avanza hasta la posición actual del término de la derecha y verifica si también terminó.
- En caso de no haber terminado significa que el término de la izquierda era menor, entonces en q_8 , q_9 y q_{10} , limpia el término de la derecha y decodifica el de la izquierda.
- Por otro lado si también termina el término de la derecha, significa que eran de igual tamaño, donde avanza a q_{11} y de aquí se devuelve al comienzo de la cinta.
- Acá para comparar son dos partes virtualmente simétricas, dependiendo si comienza con 0 o 1 (codificado), lo decodifica y avanza al término de la derecha para comparar hasta q_{16} y q_{17} respectivamente.

- En caso de encontrar el mismo valor, se devuelve a q12 para seguir comparando.
- De no ser así se sale en q19 y q22 para terminar devolviendo el término adecuado en el resto de los nodos de esta rama.
- Finalmente en la situación de que ambos sean completamente iguales, se sale de q12 a q13, estando ya decodificados por el loop los términos, simplemente borra el de la derecha.

2. Pregunta 2



La maquina de Turing presenta tres cintas, la primera es aquella que presenta el input w y las otras dos son cintas inicialmente vacías.

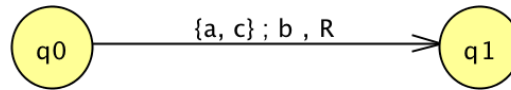
El funcionamiento de la maquina es sencillo, y al igual que el anterior se divide en dos partes:

- El loop presente en la maquina (q_0, q_1, q_2, q_3) refiere a leer la primera palabra de w y escribirla en la segunda cinta, dejándola como "X" en la primera, luego se dirige al final de la primera cinta y lee el último termino para escribirlo en la tercera cinta dejando el valor de la primera como "X" nuevamente. Realiza el mismo proceso con la segunda y penúltima letra y así sucesivamente. Si la cantidad de letras es impar no logra entrar a la segunda parte, en cambio si la cantidad es par sí entra.
- Cuando la cantidad de letras es par entra en esta sección (q_7), lo primero que hace es llevar la segunda cinta al inicio para luego empezar a comparar la segunda y tercera cinta, como la tercera cinta se empezó escribiendo de atrás para adelante se compara la segunda cinta desde el inicio moviéndose hacia la derecha y la tercera desde el final moviéndose a la izquierda, si la comparación es fructífera llega al estado de aceptación, si no lo es, no lo hará.

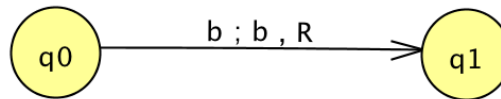


3. Pregunta 3

Vamos a demostrar que hay equivalencia entre las MT comunes y las que incluyen las transiciones creadas por Juan Turing Contreras, sólo bastaría en lugar de pasar el input requerido para la transición, utilizar el complemento de él, donde por letra se realiza la misma transición, lo que podemos ver fácilmente en el ejemplo entregado: Con $L = \{a,b,c\}$.



Donde la transición simplemente pasaría a ser el complemento del input $\{a,c\}$ en MT:



Acá también es fácil ver que nuevamente para hacer el camino hacia el otro lado es de igual forma, utilizando el complemento de $\{b\}$, por lo que notamos que ésto se puede realizar para cualquier transición que haya, por lo que generalizando lo antes dicho:

Siendo L_1 el conjunto de letras que se incluyen en la transición de la nueva MT, su transición correspondiente sería $\delta(q_x, L_1) = (q_y, v, Mov)$, donde v es por lo que se quiere reemplazar y Mov el lado hacia el que se quiere mover, entonces en MT comunes correspondería al conjunto de transiciones de la forma $\delta(q_x, u) = (q_y, v, Mov), \forall u \in L_1^c$.

Por otro lado si se tiene una transición de una MT común $\delta(q_x, u) = (q_y, v, Mov), u \in L_2$ se traduce a la nueva MT como $\delta(q_x, L_2 - \{u\}) = (q_y, v, Mov)$, por lo que queda entonces demostrado.

4. Pregunta 4

4.1. Parte a

Elijamos $w = a^{n^2}b^nc^n \in L_1$, con $|vy| > 0$ y $|vxy| < n$

Sea w de la forma $w = uvxyz$, al bombear encontramos con 5 casos:

1) "u" e "y" se encuentren dentro de a^{n^2} .



- 2) "u" e "y" se encuentren dentro de b^n .
- 3) "u" e "y" se encuentren dentro de c^n .
- 4) "u" e "y" se encuentren en la intersección de a^{n^2} y b^n .
- 5) "u" e "y" se encuentren en la intersección de b^n y c^n .

Bombeamos por casos.

- 1) uvxyz se encuentran dentro de a^{n^2} .

$$u = a^p$$

$$v = a^q$$

$$x = a^r$$

$$y = a^s$$

$$z = a^{n^2-p-q-r-s}b^n c^n$$

k veces "v" e "y", quedando:

$$a^{n^2+(q+s)(k-1)}b^n c^n$$

eligiendo k=2

$$a^{n^2+(q+s)}b^n c^n$$

Para que pertenesca al lenguaje,

$$n^2 + q + s = n \cdot n$$

$$(q+s) = 0$$

Pero $|vy| > 0$, existiendo así una contradicción.

- 2) Similar al caso 1, sólo que esta vez se bombea b^n , aumentando la cantidad de estas, quedando:

$$a^{n^2}b^{n+k}c^n$$

con cualquier $k > 1$ nos escapamos del lenguaje.

- 3) Simétrico al caso 2, sólo que esta vez se bombea c^n .

- 4) "x" e "y" se encontrarán dentro de las a's y las b's, quedando algo del estilo:

$$a^{n^2+k}b^{n+k}c^n$$

bombeando hacia arriba con k=2 queda:

$$a^{n^2+2}b^{n+2}c^n$$

Como no se cumple la igualdad, nos escapamos del lenguaje.

- 5) "x" e "y" se encontrarán dentro de las b's y las c's, quedando algo del estilo:

$$a^{n^2}b^{n+k}c^{n+k}$$

bombeando hacia arriba con k=2 queda:

$$a^{n^2}b^{n+2}c^{n+2}$$

Como no se cumple la igualdad, nos escapamos del lenguaje.

Como nos escapamos en todos los casos posibles, L_1 no es LLC

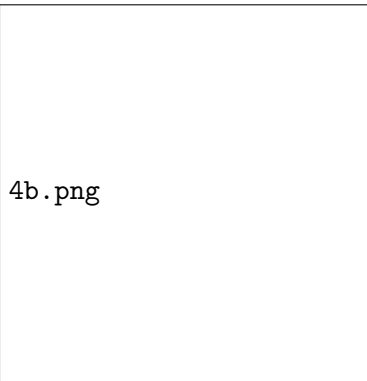


4.2. Parte b

Primero vemos la ecuación de la forma $n = 8 - \frac{5}{3}m$, donde notamos que en los enteros no negativos solo hay dos combinaciones posibles:

- $m = 0; n = 8$
- $m = 3; n = 3$

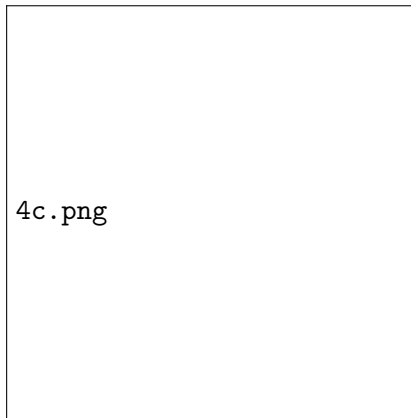
Por lo que a continuación solo basta hacer el autómata correspondiente:



Acá verificamos que incluso el lenguaje es regular, por lo que es de libre contexto.

4.3. Parte c

Al igual que antes, es más fácil armar el PDA viendo la ecuación de otra forma: $5m = 24 + 3n$



Como hay PDA que lo represente, entonces es LLC.

4.4. Parte d

A continuación se muestra el PDA del lenguaje, y como tiene PDA, entonces es LLC.



4d.png

4.5. Parte e

Elijamos $w = a^n b^n c a^n b^n \in L_5$, con $|vy| > 0$ y $|vxy| < n$

Sea w de la forma $w = uvxyz$, al bombear nos encontramos con 7 casos:

- 1) "u" e "y" se encuentren dentro de las primera a^n .
- 2) "u" e "y" se encuentren dentro de las segundas a^n .
- 3) "u" e "y" se encuentren dentro de las primeras b^n .
- 4) "u" e "y" se encuentren dentro de las segundas b^n .
- 5) "u" e "y" se encuentren en la primera intersección de a's y b's.
- 6) "u" e "y" se encuentren en la segunda intersección de a's y b's.
- 7) "u" e "y" se encuentren en la intersección de las a^n con la c.

Bombeamos por casos.

- 1) $uvxyz$ se encuentran dentro de a^n .

$$u = a^p$$

$$v = a^q$$

$$x = a^r$$

$$y = a^s$$

$$z = a^{n-p-q-r-s} b^n c a^n b^n$$

k veces "v" e "y", quedando:

$$a^{n+(q+s)(k-1)} b^n c a^n b^n$$

eligiendo $k=0$

$$a^{n-q-s} b^n c a^n b^n$$

Para que pertenesca al lenguaje,

$$n-q-s+n = n+n$$

$$-(q+s) = 0$$

Pero $|vy| > 0$, existiendo así una contradicción.

- 2) Simétrico al caso 1, sólo que esta vez se bombea las segundas a^n .



- 3) Simétrico al caso 1, sólo que esta vez se bombean las primeras b^n .
- 4) Simétrico al caso 1, sólo que esta vez se bombean las segundas b^n .
- 5) $|vxy|$ se encuentra entre $a^n b^n$, siguiendo la lógica del caso 1) es evidente que si bombeamos hacia arriba aumentamos los valores de a y b, y no se cumpliría la igualdad, ya que los primeros a's y b's son más que los segundos, escapando así del lenguaje.
- 6) Simétrico al caso 5, sólo que esta vez se bombean las segundas $a^n b^n$.
- 7) En los casos en que c se encuentre dentro de "v" o "y" es evidente que al bombear hacia arriba va a aumentar la cantidad de c en la palabra y se escapa logicamente del lenguaje, entonces el caso a analizar es cuando c se encuentra en x.

Así, si bombeamos hacia abajo, se eliminan las b's anteriores a c y las a's seguidas a c.

$$a^n b^{n-k} c a^{n-k} b^n$$

$k < 0$ nos escapamos del lenguaje.

Como nos escapamos en todos los casos posibles, L_5 no es LLC