

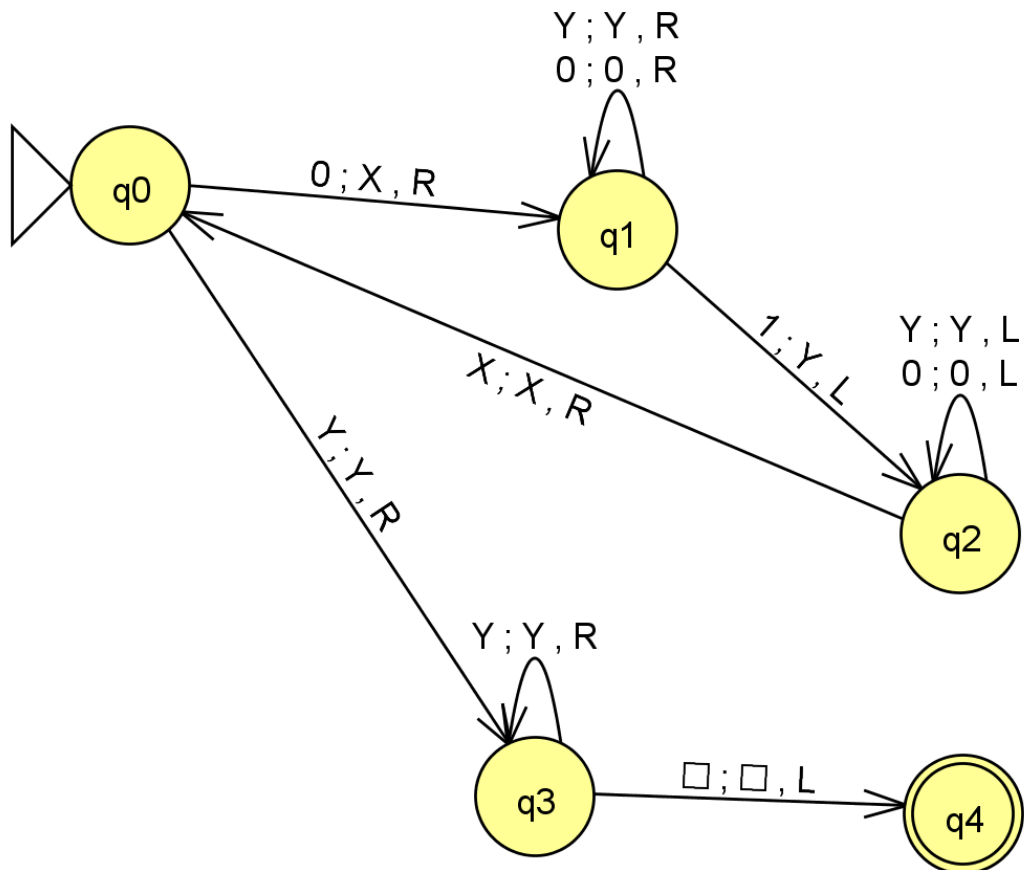
INFORME DE LA PRÁCTICA 9

Nº 1.- Dada la siguiente máquina de Turing, comprobar con el JFLAP paso a paso que con el lenguaje $L = \{0^n 1^n \mid n \geq 1\}$ se acepta tres cadenas y otras tres no.

Esta es la Máquina de Turing:

δ	0	1	X	Y	\$
q_0	(q_1, X, R)	-	-	(q_3, Y, R)	-
q_1	$(q_1, 0, R)$	(q_2, Y, L)	-	(q_1, Y, R)	-
q_2	$(q_2, 0, L)$	-	(q_0, X, R)	(q_2, Y, L)	-
q_3	-	-	-	(q_3, Y, R)	$(q_4, \$, L)$
q_4	-	-	-	-	-

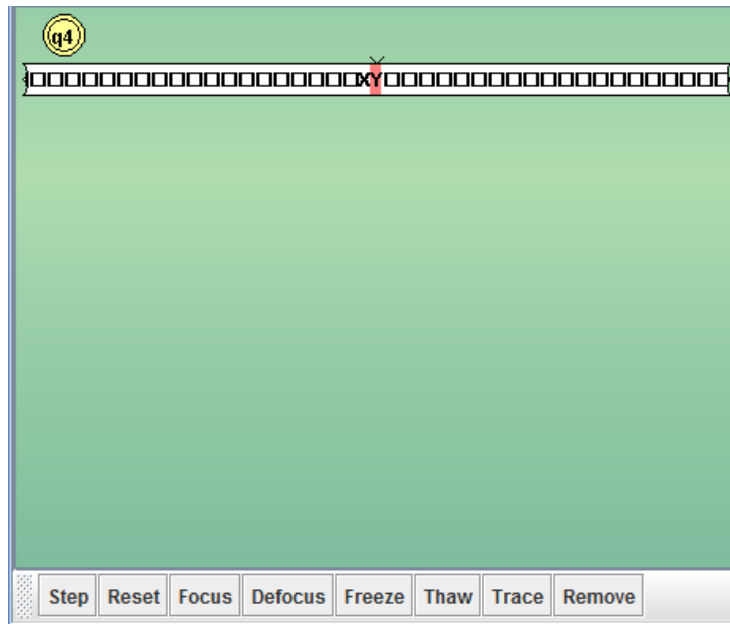
Así quedaría implementado en el JFLAP:



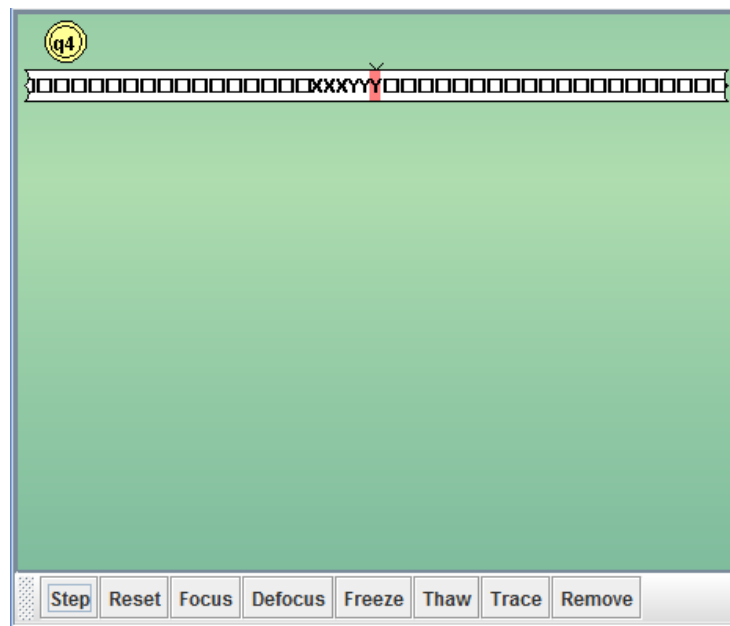
Escuela Superior de Ingeniería y Tecnología. Sección de Ingeniería Informática

Analizamos tres cadenas que si acepte:

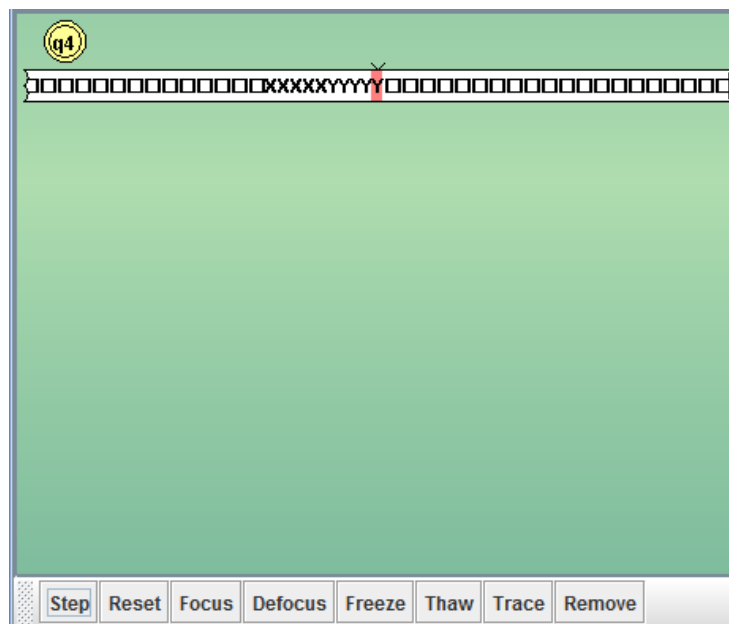
1. *01:*



2. 000111 :

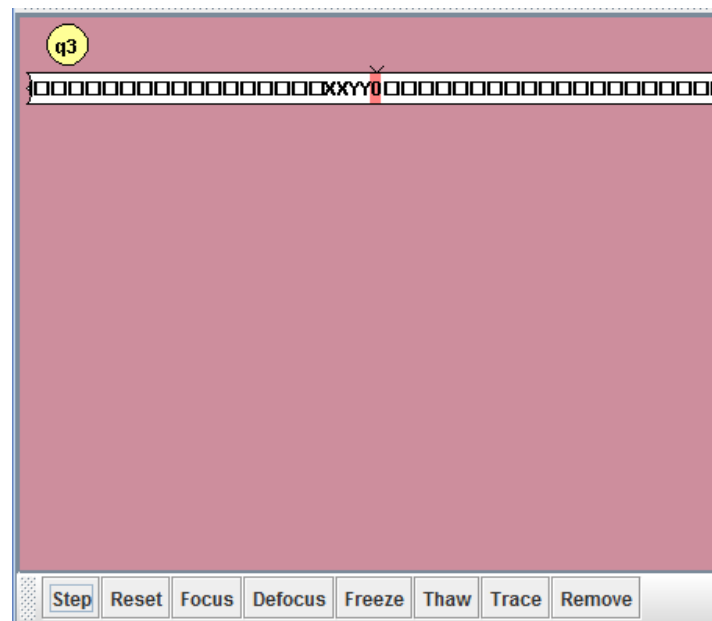


3. 0000011111:

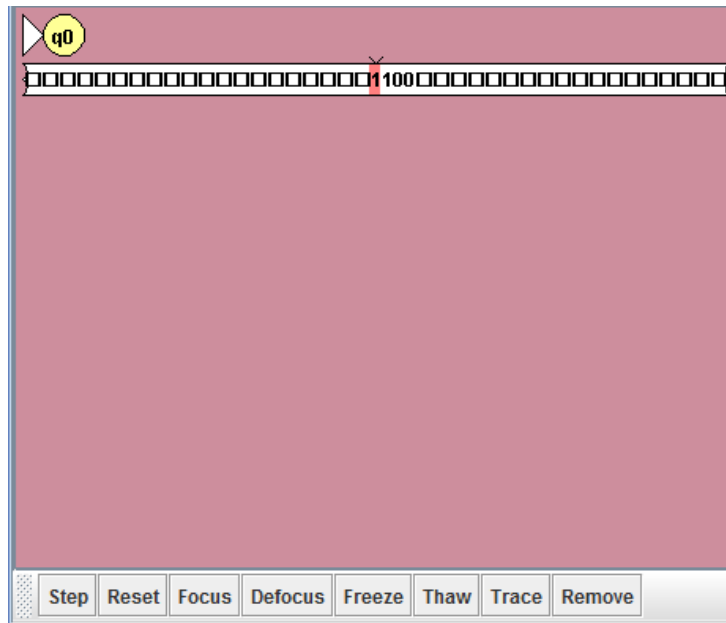


Ahora analizamos tres cadenas que **NO** acepte:

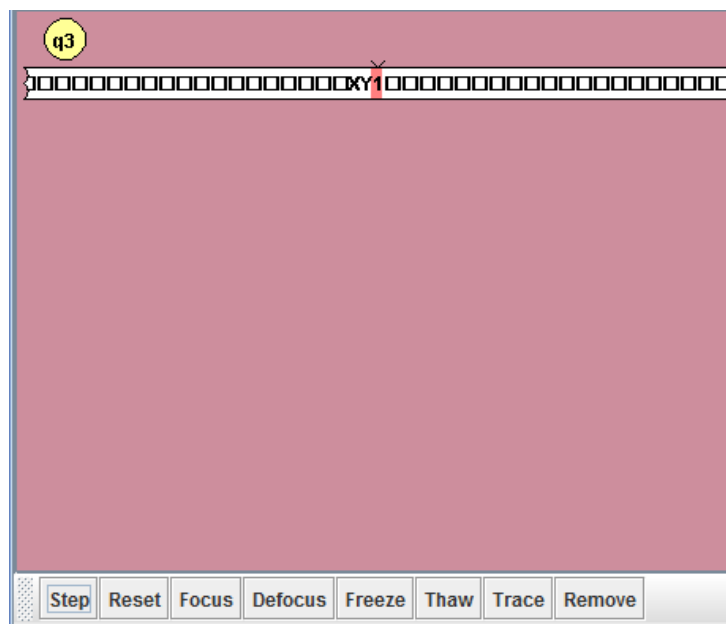
1. 00110:



2. 1100:



3. 011:

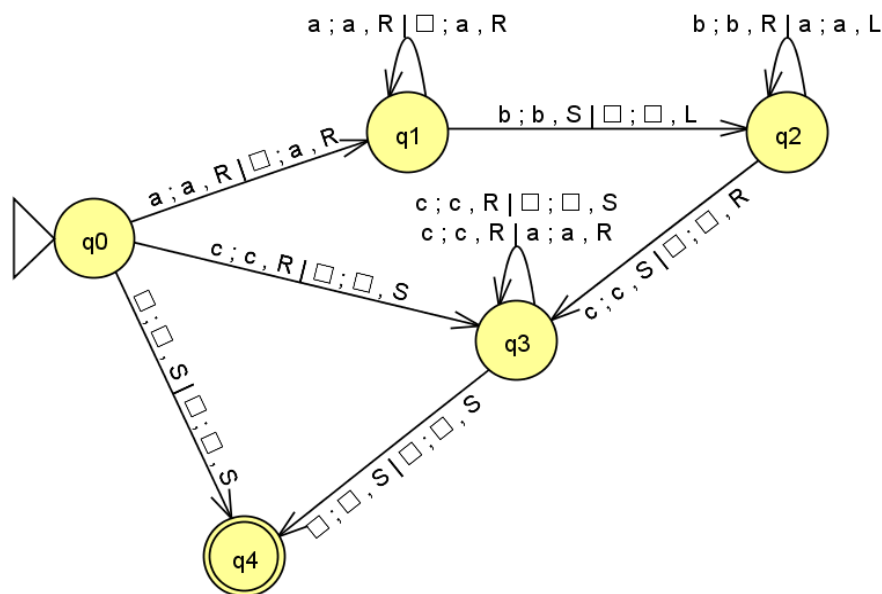


Nº 2.- Se nos pide implementar una máquina de Turing que reconozca el lenguaje $L = \{a^n b^n c^m \mid n \geq 0, m \geq n\}$. Además, debemos comprobar nuestra máquina para una cadena que si sea aceptada y una para la que no.

Nuestra máquina quedaría así:

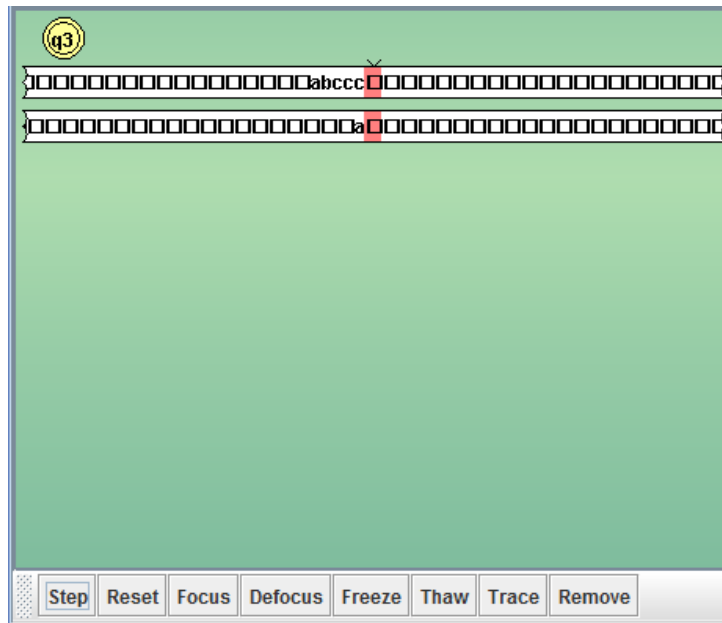
$$\begin{aligned} \delta(q_0, (a, \text{b})) &\rightarrow (q_1, (a, a), (R, R)) \\ \delta(q_0, (c, \text{b})) &\rightarrow (q_3, (c, \text{b}), (R, S)) \\ \delta(q_0, (\text{b}, \text{b})) &\rightarrow (q_4, (\text{b}, \text{b}), (S, S)) \\ \delta(q_1, (a, \text{b})) &\rightarrow (q_1, (a, a), (R, R)) \\ \delta(q_1, (b, \text{b})) &\rightarrow (q_2, (b, \text{b}), (S, L)) \\ \delta(q_2, (b, a)) &\rightarrow (q_2, (b, a), (R, L)) \\ \delta(q_2, (c, \text{b})) &\rightarrow (q_3, (c, \text{b}), (S, R)) \\ \delta(q_3, (c, \text{b})) &\rightarrow (q_3, (c, \text{b}), (R, S)) \\ \delta(q_3, (c, a)) &\rightarrow (q_3, (c, a), (R, R)) \\ \delta(q_3, (\text{b}, \text{b})) &\rightarrow (q_4, (\text{b}, \text{b}), (S, S)) \end{aligned}$$

En JFLAP quedaría implementado de la siguiente manera. Utilizando dos cintas, ponemos la cadena en la primera, y en la segunda la dejamos en blanco. Se empiezan reconociendo aes o ces o la cadena vacía. Si nos aparece a, apuntamos todas las que salgan en la segunda cinta. Después solo pueden venir bes, por lo que analizamos la segunda cinta moviéndonos a la izquierda y la primera a la derecha, para comprobar que tengan el mismo número. Después de las bes, solo pueden venir ces, por lo que ahora volvemos a analizar que, al menos, tengan la misma cantidad de aes por lo que movemos ambas cintas a la derecha. También analizamos que ahora pueden venir tantas ces como quieran, pero no pueden venir ninguna a ni b.



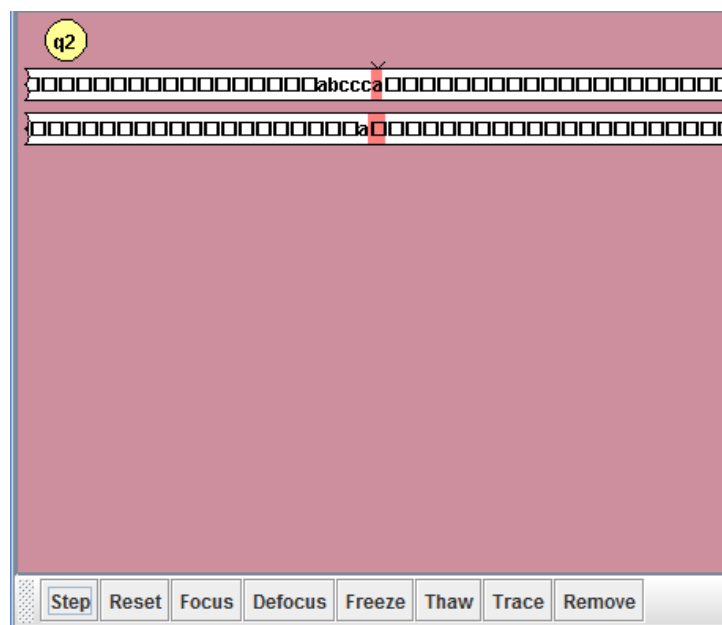
Una cadena que debería aceptar esta máquina de Turing sería:

1. *abccc*:



Una cadena que no debería aceptar sería:

1. *abccca*:

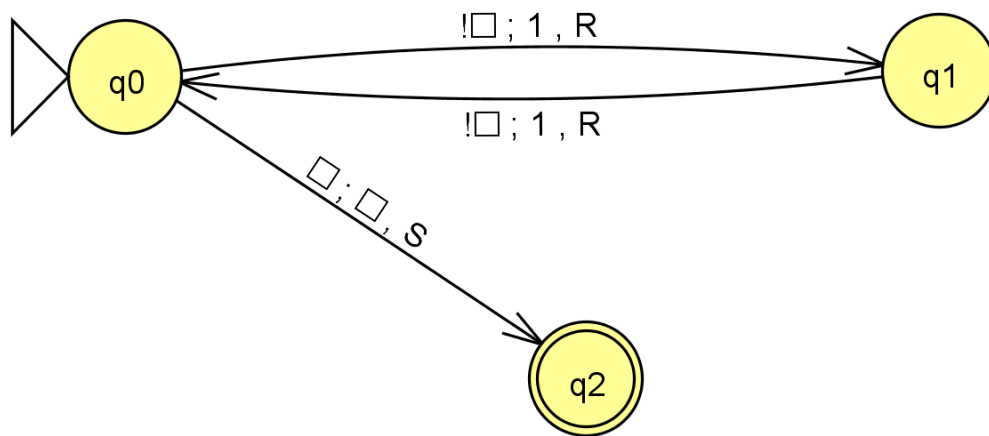


Nº 3.- Teniendo el siguiente lenguaje $L = \{w \mid \text{la longitud de } w \text{ es par}\}$, implementar una máquina de Turing que reconozca dicho lenguaje y analizar una cadena que sea aceptada y otra que no.

La máquina de Turing nos quedaría de la siguiente manera:

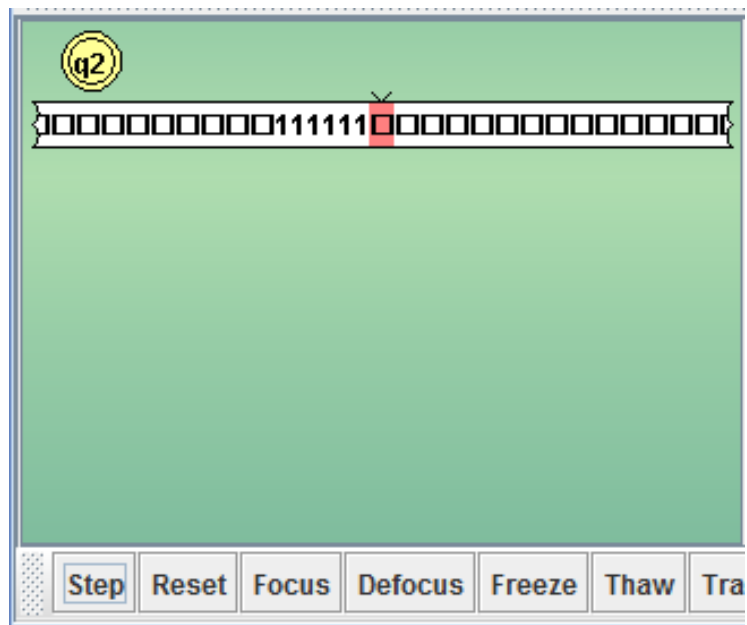
δ	w	\$
q_0	(q_1, w, R)	$(q_2, \$, S)$
q_1	(q_0, w, R)	—
q_2	—	—

En el JFLAP tendríamos el siguiente esquema. Si no nos sale una cadena (la cadena vacía), se acepta, ya que su longitud es par. Si nos sale cualquier carácter, pasamos a un nuevo estado, estado que representaría las cadenas de longitud impar y nos movemos a la derecha. Si nos vuelve a salir cualquier carácter tenemos que aceptarla, por lo que volvemos al estado inicial y nos movemos a la derecha. Si solo salen blancos, ya tenemos longitud par.



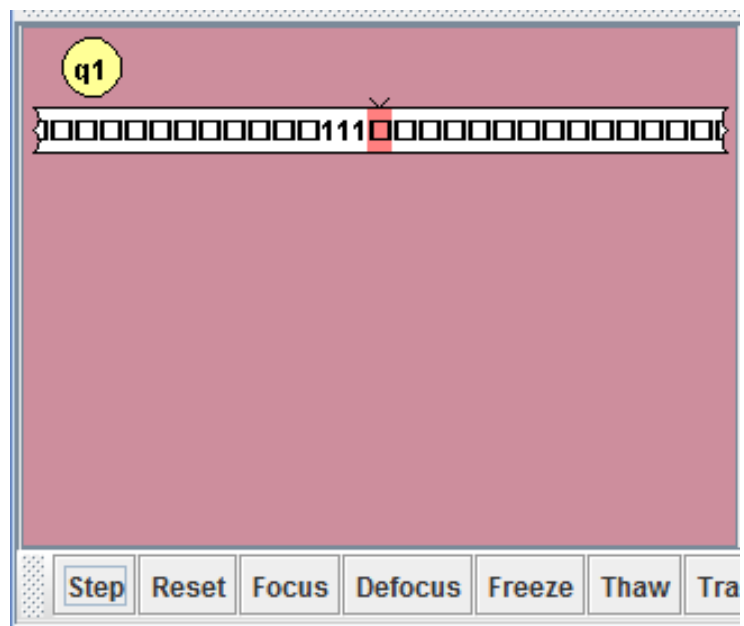
Analizamos una cadena que si debe ser aceptada:

1. 123456:



Ahora analizamos una cadena que **NO** debe de ser aceptada:

2. zxy:



MODIFICACIÓN DE LA PRÁCTICA:

1. Diseñar una máquina de Turing que represente el siguiente lenguaje:
 $L = \{a^n b^n c^m | n = 2 * m; n \geq 0\}$.

Esta sería nuestra máquina de Turing. Es una modificación de la máquina del segundo ejercicio, pero ahora de entrada solo se puede aceptar la cadena vacía y aquellas que empiecen por aes. Nos movemos a la derecha en ambas cintas y escribimos por cada dos aes que salgan una. Lo mismo con las bes, por cada dos que salgan en la segunda cinta nos movemos una sola vez cada dos bes. Ahora solo pueden salir ces, por lo que para que la cadena se acepte, tienen que ser la mitad de aes, así que por cada c, nos movemos una sola vez. Nuestra máquina también detecta que el número de aes o de bes sea par, para que podamos tener la mitad de ces.

