TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES

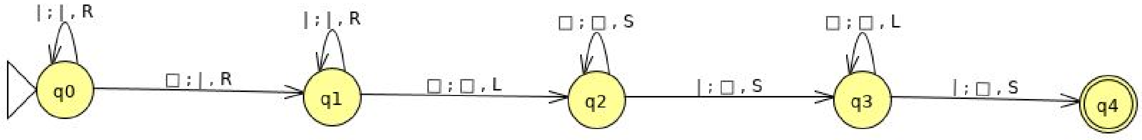
PRÁCTICA 3

EJERCICIO 1. Define the TM solution of exercise 3.4 of the problem list and test its correct behaviour.

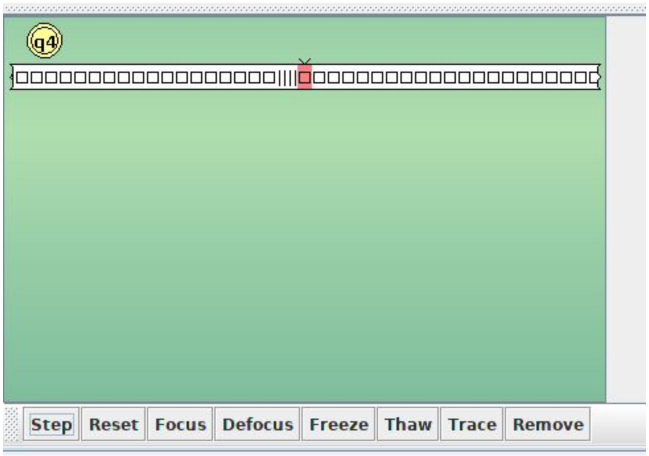
La máquina de Turing solución del ejercicio 3.4 es la siguiente:

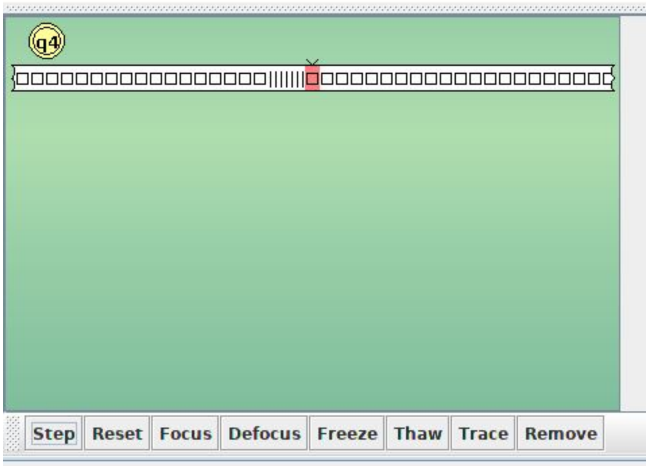
$$\begin{bmatrix} 0 & * & l & 1 \\ 0 & | & | & 0 \\ 1 & * & | & 2 \\ 1 & | & l & 1 \\ 2 & * & l & 3 \\ 2 & | & r & 2 \\ 3 & * & l & 4 \\ 3 & | & * & 3 \\ 4 & * & h & 4 \\ 4 & | & * & 4 \end{bmatrix}$$

Haciendo uso de JFLAP (teniendo en cuenta que la máquina de Turing en el mismo comienza con el cabezal a la izquierda de las cadenas, al contrario que la planteada en nuestro ejercicio en clase, en el cual se comienza a la derecha de las mismas) quedaría de la forma:



Para probarlo, hemos introducido las cadenas $| * |||$ y $||| * |||$, es decir, intentaremos sumar $0 + 3$ y $3 + 3$ (recordemos que estamos trabajando con una representación unaria). Efectivamente, se obtienen para el primer caso, la cadena $|||$ que representa al número 3, mientras que en el segundo caso, la cadena $||||||$, representando al número 6, tal como queríamos probar:





EJERCICIO 2. Define a recursive function for the sum of three values.

Según la definición siguiendo los apuntes, si $k \geq 0$ y tenemos las funciones

$g : \mathbb{N}^k \to \mathbb{N}$

$h : \mathbb{N}^{k+2} \to \mathbb{N}$

Si la función $f : \mathbb{N}^{k+1} \to \mathbb{N}$ es

$$f(\vec{n}, m) = \begin{cases} g(\vec{n}) & \text{if} \quad m = 0 \\ h(\vec{n}, m-1, f(\vec{n}, m-1)) & \text{if} \quad m > 0 \end{cases}$$

entonces f se obtiene de g y h por recursión primitiva.

Sabemos también por que hemos visto en clase, que $suma(n) = \quad < \pi_1^1 | \sigma(\pi_3^3) > (n)$

Ahora la función $suma_3$ quería de la forma:

$suma_3$: $\mathbb{N}^3 \rightarrow \mathbb{N}$

Vamos a tomar entonces g y h como:

$g : \mathbb{N}^2 \rightarrow \mathbb{N}, g \equiv suma(x, y)$

$h : \mathbb{N}^4 \rightarrow \mathbb{N}, h \equiv \sigma(\pi_4^4)$

Así nuestra función estaría definida como:

$suma_3(n) =<< \pi_1^1 \;\mid\; \sigma(\pi_3^3) > \;\mid\; \sigma(\pi_4^4) > (n)$

Probemos ahora en Octave, sumando 1,4 y 3, para lo que el programa deberá devolvernos como solución, 8:

```
octave:2> evalrecfunction('suma_3',1,4,3)
suma₃(1,4,3)
<<π¹₁|σ(π³₃)>|σ(π⁴₄)>(1,4,3)
<<π¹₁|σ(π³₃)>|σ(π⁴₄)>(1,4,2)
<<π¹₁|σ(π³₃)>|σ(π⁴₄)>(1,4,1)
<<π¹₁|σ(π³₃)>|σ(π⁴₄)>(1,4,0)
<π¹₁|σ(π³₃)>(1,4)
<π¹₁|σ(π³₃)>(1,3)
<π¹₁|σ(π³₃)>(1,2)
<π¹₁|σ(π³₃)>(1,1)
<π¹₁|σ(π³₃)>(1,0)
π¹₁(1) = 1
σ(π³₃)(1,0,1)
π³₃(1,0,1) = 1

σ(1) = 2
σ(π³₃)(1,1,2)
π³₃(1,1,2) = 2

σ(2) = 3
σ(π³₃)(1,2,3)
π³₃(1,2,3) = 3

σ(3) = 4
σ(π³₃)(1,3,4)
π³₃(1,3,4) = 4

σ(4) = 5
σ(π⁴₄)(1,4,0,5)
π⁴₄(1,4,0,5) = 5

σ(5) = 6
σ(π⁴₄)(1,4,1,6)
π⁴₄(1,4,1,6) = 6

σ(6) = 7
σ(π⁴₄)(1,4,2,7)
π⁴₄(1,4,2,7) = 7

σ(7) = 8
ans =  8
```

EJERCICIO 3. Implement a WHILE program that computes the sum of three values. You must use an auxiliary variable that accumulates the result of the sum.

$[suma_3]$

$X_4 := X_1;$

while $X_2 \ != \ 0$ **do**

$X_4 := X_4 + 1;$

$X_2 := X_2 - 1;$

od

while $X_3 \ != \ 0$ **do**

$X_4 := X_4 + 1;$

$X_3 := X_3 - 1;$

od