Teoría de Autómatas y Lenguajes Formales

Práctica 3

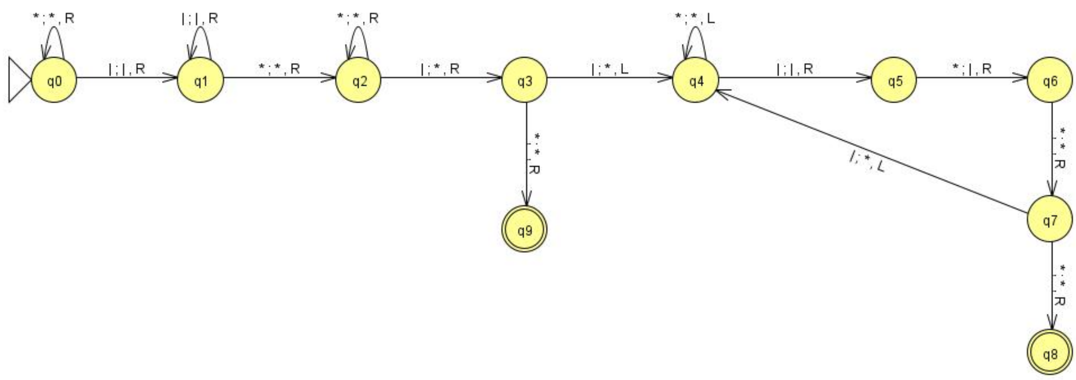
Marina González Torres

Ejercicio 1

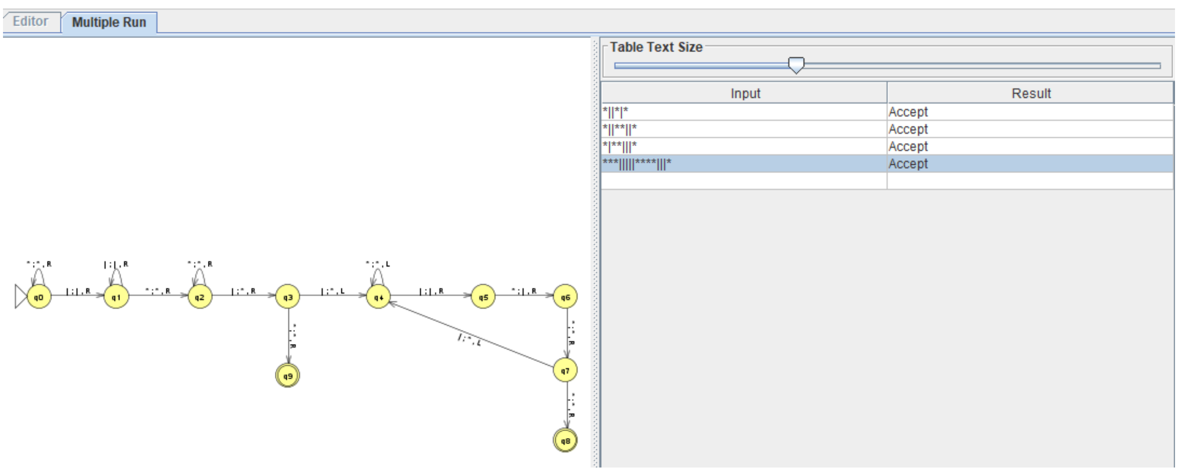
Defina la solución de la Maquina de Turing del ejercicio 3.4 de la lista de ejercicio y pruebe su correcto funcionamiento.

El ejercicio 3.4 nos dice que debemos probar que la funcion add(x, y) con x e y perteneciente a los números naturales es computable por Turing usando la notación unaria {}

La máquina de Turing simulada con el JFlap quedaría de la siguiente forma:



Para poder visualizar el correcto funcionamiento adjunto un ejemplo de ejecución múltiple:



Ejercicio 2

Defina una función recursiva para la suma de tres valores.

Para realizar este ejercicio hemos usado unos ficheros existentes en el repositorio de ”talfuma” llamados ”recursivefunctions” y ”evalrecfunction”.
En el primer fichero hemos añadido una funcion a la que hemos llamado addition2 que realiza la función recursiva de tres valores.

```
Abrir ▾                    recursivefunctions
                   ~/tafluma/software/recursivefunctions
constant^1_0         <θ|π^2_2>
constant^1_1         σ(<θ|π^2_2>)
constant^2_3         <<σ(σ(σ(θ)))|π^2_2>|π^3_3>
addition             <π^1_1|σ(π^3_3)>
addition2            <<π^1_1|σ(π^3_3)>|σ(π^4_4)>
predecessor          <θ|π^2_1>
subtraction          <π^1_1|predecessor(π^3_3)>
product              <constant^1_0|<π^1_1|σ(π^3_3)>(π^3_1,π^3_3)>
division             μ[subtraction(π^3_1,product(π^3_2,π^3_3))]
power                <constant^1_1|product(π^3_1,π^3_3)>
squareroot           μ[subtraction(π^2_1,product(π^2_2,π^2_2))]
cuberoot             μ[subtraction(π^2_1,power(π^2_2,constant^2_3))]
```

Luego hemos probado su funcionamiento con Octave y un ejemplo de ello sería el siguiente:

```
octave:1> evalrecfunction('addition2',1,2,3)
addition2(1,2,3)
<<π¹₁|σ(π³₃)>|σ(π⁴₄)>(1,2,3)
<<π¹₁|σ(π³₃)>|σ(π⁴₄)>(1,2,2)
<<π¹₁|σ(π³₃)>|σ(π⁴₄)>(1,2,1)
<<π¹₁|σ(π³₃)>|σ(π⁴₄)>(1,2,0)
<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(π⁴₄)(1,2,0,3)
π⁴₄(1,2,0,3) = 3

σ(3) = 4
σ(π⁴₄)(1,2,1,4)
π⁴₄(1,2,1,4) = 4

σ(4) = 5
σ(π⁴₄)(1,2,2,5)
π⁴₄(1,2,2,5) = 5

σ(5) = 6
ans =  6
```

Ejercicio 3

Implemente un programa WHILE que calcule la suma de tres valores. Debe utilizar una variable auxiliar que acumule el resultado de la suma.

La función WHILE que implementa la suma de los tres valores quedaría de la siguiente manera:

```
3,
X4:=X1;
while X2!=0 do
        X4:=X4+1;
        X2:=X2-1;
        while X3!=0 do
        X4:=X4+1;
        X3:=X3-1
        od
od;
X1:=X4)
```

Ejemplo de ejecución:

```
octave:1> F("(3, X4:=X1; while X2!=0 do X4:=X4+1; X2:=X2-1; while X3!=0 do X4:=X
4+1; X3:=X3-1 od od; X1:=X4)",1,2,3)
X1 =  6
ans =  6
```