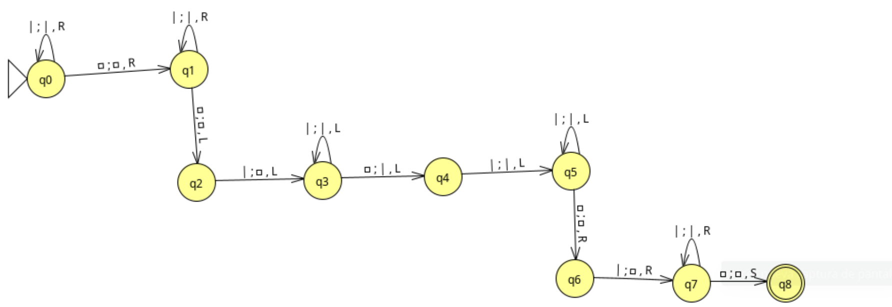
Práctica 3

Juan Miguel Fernández Tejada

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



Debido a que el ejercicio nos pide que la máquina de Turing inicie su recorrido al final de la cadena y en JFLAP inicia al principio, son necesarios los dos primeros estados para llevar la máquina al lugar requerido. Posteriormente empleamos los cuatro estados siguientes para eliminar el último | y desplazar a la izquierda la máquina, sustituyendo el espacio entre los números (grupos de |) por un |. Dado que cada número se construye con $n + 1$ símbolos, al sumar dos números n y m tendremos $n + m + 2$ símbolos, con lo cual nos sigue sobrando uno. Por eso una vez se alcanza el extremo izquierdo de la secuencia de símbolos, se cambia el primero por otro caracter vacío y se vuelve a desplazar la máquina hasta el final de la cadena, que es donde debe acabar; de esto se encargan los últimos tres estados.

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

Sabemos que una función recursiva de tres argumentos tendrá esta estructura:

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

A partir de esto, la propuesta es la siguiente:

$$< (< \pi_1^1|\sigma(\pi_3^3) >)|\sigma(\pi_4^4) >$$

Comprobemos que esta función realiza efectivamente la suma de tres valores.

$$f(n,m,p) = \begin{cases} < \pi_1^1 | \sigma(\pi_3^3) > (n,m) & p = 0 \\ \sigma(\pi_4^4)(n,m,p-1,f(n,m,p-1)) & p > 0 \end{cases}$$

Desarrollemos primero la función correspondiente a g:

$$g(n,m) = \begin{cases} \pi_1^1(n) & m = 0 \\ \sigma(\pi_3^3)(n,m-1,g(n,m-1)) & m > 0 \end{cases}$$

$$\begin{aligned} \sigma(\pi_3^3)(n,m-1,g(n,m-1)) &= g(n,m-1)+1 = g(n,m-2)+2 = \\ &= ... = g(n,0)+m = \pi_1^1(n)+m = n+m \end{aligned}$$

Hagamos lo mismo con la función correspondiente a h:

$$\begin{aligned} h(n,m,p-1,f(n,m,p-1)) &= \sigma(\pi_4^4)(n,m,p-1,f(n,m,p-1)) = f(n,m,p-1)+1 = \\ &= f(n,m,p-2)+2 = ... = f(n,m,0)+p = \\ &=< \pi_1^1 | \sigma(\pi_3^3) > (n,m)+p = n+m+p \end{aligned}$$

Con lo cual queda demostrado que la función propuesta realiza lo que se pide. En Octave:

```
octave:18> evalrecfunction("<<pi^1_1|sigma(pi^3_3)>|sigma(pi^4_4)>", 1, 2, 3)
<<π^1_1|σ(π^3_3)>|σ(π^4_4)>(1,2,3)
<<π^1_1|σ(π^3_3)>|σ(π^4_4)>(1,2,2)
<<π^1_1|σ(π^3_3)>|σ(π^4_4)>(1,2,1)
<<π^1_1|σ(π^3_3)>|σ(π^4_4)>(1,2,0)
<π^1_1|σ(π^3_3)>(1,2)
<π^1_1|σ(π^3_3)>(1,1)
<π^1_1|σ(π^3_3)>(1,0)
π^1_1(1) = 1
σ(π^3_3)(1,0,1)
π^3_3(1,0,1) = 1

σ(1) = 2
σ(π^3_3)(1,1,2)
π^3_3(1,1,2) = 2

σ(2) = 3
σ(π^4_4)(1,2,0,3)
π^4_4(1,2,0,3) = 3

σ(3) = 4
σ(π^4_4)(1,2,1,4)
π^4_4(1,2,1,4) = 4

σ(4) = 5
σ(π^4_4)(1,2,2,5)
π^4_4(1,2,2,5) = 5

σ(5) = 6
ans = 6
```

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.

Q = (3, s)

s:

```
X_4 := 0
while X_3 ≠ 0 do
    X_4 := X_4 + 1
    X_3 := X_3 − 1
od
while X_2 ≠ 0 do
    X_4 := X_4 + 1
    X_2 := X_2 − 1
od
while X_1 ≠ 0 do
    X_4 := X_4 + 1
    X_1 := X_1 − 1
od
X_1 := X_4
```