Práctica 3

Jesús Fuentes Moya

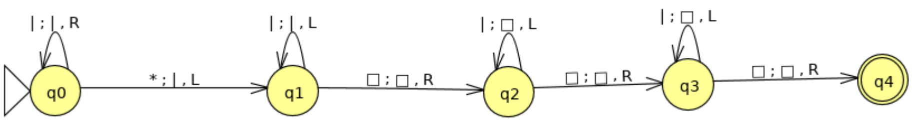
18 de noviembre de 2022

Índice general

Capítulo 1

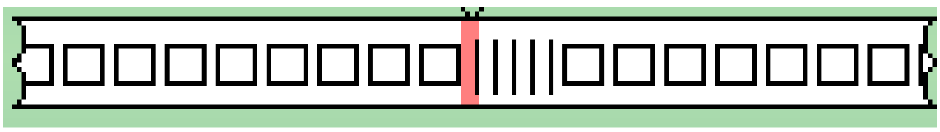
Máquina de Turing definida con JFLAP

La máquina de Turing que realiza la operación suma es:



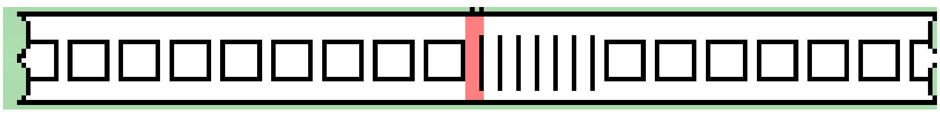
Veamos que la máquina de Turing descrita funciona correctamente. Para ello probemos un par de ejemplos:

Ejemplo 1. Introducir las cadenas: $|| \ast |||$ y el resultado ha de ser $||||$



Como podemos observar la maquina de Turing nos devuelve $||||$ por tanto funciona correctamente

Ejemplo 2. Introducir las cadenas: $||||| \ast ||$ y el resultado ha de ser $|||||||$



En este ejemplo la maquina de Turing vuelve a funcionar correctamente ya que nos devuelve $|||||||$.

Capítulo 2

Definir una función REC que realice la suma de tres valores

Para aplicar una función recursiva primitiva necesitamos definir una función para el caso base g y una función para el caso recursivo f, que va a tomar como parámetros

- Paso en el que estoy: $m - 1$
- Cuál es el resultado de ese paso: $h(n, m - 1)$
- Parámetros n

y devolver el valor de la función en el paso m.

Sea $K \geq 0$ y 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

$$f(n, m) = \begin{cases} g(n) & m = 0 \\ h(n, m-1, f(n, m-1)) & m > 0 \end{cases} \tag{2.1}$$

Entonces, f se obtiene de g y h por recursión primitiva.

Sabiendo esto definamos una función REC que realice la suma de tres valores, es decir, hallemos la función $suma_3$ como una función recursiva.
Sea $k = 2$ y sean las funciones
$g : \mathbb{N}^2 \to \mathbb{N}$
$h : \mathbb{N}^{2+2} \to \mathbb{N}$

$suma_3 : \mathbb{N}^{2+1} \rightarrow \mathbb{N}$

Entonces, se tiene que:
$g : \mathbb{N}^2 \rightarrow \mathbb{N}$
$h : \mathbb{N}^4 \rightarrow \mathbb{N}$
$suma_3 : \mathbb{N}^3 \rightarrow \mathbb{N}$

Tomamos $g = suma(x, y)$ y $h = \sigma(\pi_4^4)$

Veamos que, como hemos visto en teoría, *suma* es una función recursiva.
$suma(n) = \langle \pi_1^1 | \sigma(\pi_3^3) \rangle (n)$

Entonces, la función $suma_3$ queda:
$suma_3(n) = \langle \langle \pi_1^1 | \sigma(\pi_3^3) \rangle | \sigma(\pi_4^4) \rangle (n)$

Comprobemos su funcionamiento con la herramienta Octave

```
>> evalrecfunction('addition3',1,1,1)
addition3(1,1,1)
<<π^1_1|σ(π^3_3)>|σ(π^4_4)>(1,1,1)
<<π^1_1|σ(π^3_3)>|σ(π^4_4)>(1,1,0)
<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^4_4)(1,1,0,2)
π^4_4(1,1,0,2) = 2

σ(2) = 3
ans =  3
```

Como podemos observar la función recursiva funciona perfectamente ya que $1 + 1 + 1 = 3$.

Capítulo 3

Definir programa WHILE que realice la suma de tres valores

while $X_2 \neq 0$ **do**
$\quad X_1 := X_1 + 1;$
$\quad X_2 := X_2 - 1$
od
while $X_3 \neq 0$ **do**
$\quad X_1 := X_1 + 1;$
$\quad X_3 := X_3 - 1$
od