Atividade III

Aluno: Caetano Colin Torres

7. A codificação church é usada para incorporar dados e operadores ao cálculo lambda:

Número 0: λf.λx. x
Número 1: λf.λx. f x
. . .
Número N: λf.λx. F*n* x

Função Sucessor: λn.λf.λx. f (n f x)

Função Predecessor:

$$\mathrm{pred}(n) = \begin{cases} 0 & \text{se } n = 0, \\ n - 1 & \text{caso contrário} \end{cases}$$

pred(n) = λn.λf.λx. n (λg.λh. h (g f)) (λu. x) (λu. u)

Soma/Subtração:
Subtração: sub ≡ `λm.λn. (n` **`pred`** `) m`
`Adição:plus` `≡ λm.λn.λf.λx. m f (n f x)`

Exemplos:
sucessor(0) = λf.λx. f (0 f x)
= λf.λx. f x
sucessor(1) = λ1.λf.λx. f (1 f x)
= λf.λx. f (f x)
0 + 1 = plus(0,1) = λ0.λ1.λf.λx. 0 f (1 f x)
= λf.λx. (f x)
1 + 2 = plus(1,2) = λ1.λ2.λf.λx 1 f (2 f x)
= λf.λx f (f (f x))

0 - λf.λx. x
1 - λf.λx. f x
2 - λf.λx. f (f x)
3 – λf.λx. f (f (f x))
4 – λf.λx. f (f (f (f x)))
5 – λf.λx. f (f (f (f (f x))))
6 – λf.λx. f (f (f (f (f (f x)))))
7 – λf.λx. f (f (f (f (f (f (f x))))))
8 – λf.λx. f (f (f (f (f (f (f (f x)))))))
9 – λf.λx. f (f (f (f (f (f (f (f (f x))))))))
10 – λf.λx. f (f (f (f (f (f (f (f (f (f x)))))))))

1 – 0: sub(1,0) = (0 pred) 1 = λf.λx. f x
2 – 1: sub(2,1) = (1 pred) 2 = λf.λx. f x

8.
A implementação do "Combinador Y" é dada por:

$$Y = \lambda f.\,(\lambda x.\,f\,(x\,x))\,(\lambda x.\,f\,(x\,x))$$

O combinador é uma função anônima e ao aplicar qualquer função, deve levar ao mesmo ponto fixo, porém o Combinador Y mostra inconsistências, o que nos leva ao "paradoxo de Curry".
O combinador é utilizado para implementar o "paradoxo de Curry", porque ele mostra que ao permitir uma função anônima chegar no zero ou muitos valores é inconsistente na lógica matemática.