Nome: Gilson Trombetta Magro

Matrícula: 18202192

Disciplina: INE5416 - Paradigmas de Programação

Turma: 04208 (2020.1)

1. QUESTÃO 7

Pesquise sobre a Codificação de Church para incorporar operadores aritméticos e números no cálculo-λ. Veja também como calcular o sucessor e predecessor de um número. Escreva algumas anotações e exemplos sobre este tópico, por exemplo, tente encontrar o sucessor de 0, 1 e 2. Também tente somar 0 + 1 e 1 + 2. Escreva os números de 1 até 10 utilizando a codificação de Church. Faça 1 - 0 e 2 - 1.

2. RESPOSTA

A Codificação de Church consiste em uma forma de representar números naturais através do cálculo-λ. Dentro dessa codificação, os números são definidos como funções de ordem superior que recebem dois parâmetros de entrada, e aplicam o primeiro sobre o segundo n vezes. Desse modo, o número zero não aplica o primeiro parâmetro nenhuma vez; o número um aplica uma vez; o dois, duas vezes, e assim por diante, conforme a tabela abaixo:

Número	**Definição**
0	$0 = \lambda x.\lambda y.y$
1	$1 = \lambda x.\lambda y.x(y)$
2	$2 = \lambda x.\lambda y.x(x(y))$
3	$3 = \lambda x.\lambda y.x(x(x(y)))$
4	$4 = \lambda x.\lambda y.x(x(x(x(y))))$
5	$5 = \lambda x.\lambda y.x(x(x(x(x(y)))))$

6	6 = λx.λy.x(x(x(x(x(x(y))))))
7	7 = λx.λy.x(x(x(x(x(x(x(y)))))))
8	8 = λx.λy.x(x(x(x(x(x(x(x(y))))))))
9	9 = λx.λy.x(x(x(x(x(x(x(x(x(y)))))))))
10	10 = λx.λy.x(x(x(x(x(x(x(x(x(x(y))))))))))
...	...

Pode-se também definir a função de sucessor como:

s = λn. λx. λy. x (n x y),

de modo que se aplicado aos números zero, um e dois, temos:

s 0 (λn. λx. λy. x (n x y)) (λw. λz. z) λx. λy. x ((λw. λz. z) x y) λx. λy. x ((λz. z) y) λx. λy. x y, que é o mesmo que 1.
s 1 (λn. λx. λy. x (n x y)) (λw. λz. w z) λx. λy. x ((λw. λz. w z) x y) λx. λy. x ((λz. x z) y) λx. λy. x (x y), que é o mesmo que 2.
s 2 (λn. λx. λy. x (n x y)) (λw. λz. w (w z)) λx. λy. x ((λw. λz. w (w z)) x y) λx. λy. x ((λz. x (x z)) y) λx. λy. x (x (x y)), que é o mesmo que 3.

Além disso, podemos definir a função de soma da seguinte maneira:

plus = λm. λn. λx. λy. m x (n x y),

e podemos aplicá-la a alguns exemplos, como a seguir:

plus 0 1 (λm. λn. λx. λy. m x (n x y)) (λw. λz. z) (λa. λb. a b) (λn. λx. λy. (λw. λz. z) x (n x y)) (λa. λb. a b) λx. λy. (λw. λz. z) x ((λa. λb. a b) x y) λx. λy. (λz. z) ((λa. λb. a b) x y) λx. λy. ((λa. λb. a b) x y) λx. λy. ((λb. x b) y) λx. λy. x y, que é o mesmo que 1.
plus 1 2 (λm. λn. λx. λy. m x (n x y)) (λw. λz. w z) (λa. λb. a (a b)) (λn. λx. λy. (λw. λz. w z) x (n x y)) (λa. λb. a (a b)) λx. λy. (λw. λz. w z) x ((λa. λb. a (a b)) x y) λx. λy. (λz. x z) ((λa. λb. a (a b)) x y) λx. λy. x ((λa. λb. a (a b)) x y) λx. λy. x ((λb. x (x b)) y) λx. λy. x (x (x y)), que é o mesmo que 3

Podemos ainda definir a função de predecessor. Esta função é um pouco mais complicada, e utiliza de uma função “container” para empacotar o valor (segundo argumento dos numerais), para que seja possível aplicar a função (primeiro argumento dos numerais) N - 1 vezes, invés de N vezes. Ela pode ser definida da seguinte forma:

pred = λn. λx. λy. n (λg. λh. h (g x)) (λu. y) (λu. u),

e abaixo temos exemplos desta função aplicada aos números um e dois.

pred 1 (λn. λx. λy. n (λg. λh. h (g x)) (λu. y) (λu. u)) (λw. λz. w z) λx. λy. (λw. λz. w z) (λg. λh. h (g x)) (λu. y) (λu. u)

λx. λy. (λz. (λg. λh. h (g x)) z) (λu. y) (λu. u) λx. λy. (λg. λh. h (g x)) (λu. y) (λu. u), e recolorindo temos: λx. λy. (λg. λh. h (g x)) (λu. y) (λu. u) λx. λy. (λh. h ((λu. y) x)) (λu. u) λx. λy. (λu. u) ((λu. y) x) λx. λy. (λu. y) x λx. λy. y, que é o mesmo que zero.
pred 2 (λn. λx. λy. n (λg. λh. h (g x)) (λu. y) (λu. u)) (λw. λz. w (w z)) λx. λy. (λw. λz. w (w z)) (λg. λh. h (g x)) (λu. y) (λu. u) λx. λy. (λz. (λg. λh. h (g x)) ((λg. λh. h (g x)) z)) (λu. y) (λu. u) λx. λy. (λg. λh. h (g x)) ((λg. λh. h (g x)) (λu. y)) (λu. u), e recolorindo, temos: λx. λy. (λg. λh. h (g x)) ((λg. λh. h (g x)) (λu. y)) (λu. u) λx. λy. (λg. λh. h (g x)) (λh. h ((λu. y) x)) (λu. u) λx. λy. (λg. λh. h (g x)) (λh. h y) (λu. u) λx. λy. (λh. h ((λh. h y) x)) (λu. u) λx. λy. (λh. h (x y)) (λu. u) λx. λy. (λu. u) (x y) λx. λy. x y, que é o mesmo que 1.

Além disso, dada a função de predecessor, podemos definir a função de subtração da seguinte forma:

minus = λm. λn. (n pred) m,

e a seguir são apresentados dois exemplos utilizando essa função:

minus 1 0 (λm. λn. (n pred) m) (λw. λz. w z) (λa. λb. b) (λn. (n pred) (λw. λz. w z)) (λa. λb. b) ((λa. λb. b) pred) (λw. λz. w z) ((λa. λb. b) pred) (λw. λz. w z)

(λb. b) (λw. λz. w z) λw. λz. w z, que é o mesmo que 1.
minus 2 1 (λm. λn. (n pred) m) (λw. λz. w (w z)) (λa. λb. a b) (λn. (n pred) (λw. λz. w (w z))) (λa. λb. a b) ((λa. λb. a b) pred) (λw. λz. w (w z)) (λb. pred b)) (λw. λz. w (w z)) pred (λw. λz. w (w z)) pred 2 1

3. REFERÊNCIAS

- https://en.wikipedia.org/wiki/Church_encoding
- http://gregfjohnson.com/pred/