Felipe Correa Lopes dos Santos
Ciências da Computação - 6º Período Noturno
Disciplina: Programação Funcional
Data: 12 ago. 2022

CURRYING

O cálculo lambda (λ) é conhecido por formar um protótipo de linguagem de programação universal, também considerada uma das “menores” linguagens de programação do mundo e na área de Tecnologia da Informação, usada em programação funcional, como em Haskell e Clean (BARENDREGT, DEKKERS, *et al*., 2013).

Ligado ao cálculo lambda, existe uma técnica chamada Currying, que possui esse nome em homenagem ao criador de seu conceito, um matemático americano chamado Haskell Curry (BARENDREGT; BARENDSEN, 2000 *apud* SILVA, 2015). Segundo Riffel, Andrade e Nogueira (2018), currying é o processo de transformar uma função de múltiplos argumentos em múltiplas funções onde cada uma possui um argumento, e as mesmas são organizadas de forma encadeada, essas ditas como funções unárias, que se relaciona com a matemática, onde uma operação unária é uma operação com somente uma variável de entrada.

No código abaixo está um exemplo da aplicação do currying utilizando a linguagem Haskell, onde é perceptível o conceito supracitado da utilização de múltiplas funções e seu encadeamento, onde cada uma possui um argumento.

$$mult :: Int \rightarrow (Int \rightarrow (Int \rightarrow Int))$$
$$mult\ x\ y\ z\ =\ x \times y \times z$$

Agora utilizando a notação lambda (λ), temos para a mesma expressão acima, que também multiplica três valores, o seguinte resultado, utilizando o exemplo da multiplicação entre 5, 7 e 3.

$(\lambda\alpha.\ (\lambda\beta.\ (\lambda\gamma.\ a\ \times b\ \times\ y)))$	$\alpha \rightarrow (\beta \rightarrow (\gamma \rightarrow x))$
$(\lambda\alpha.\ (\lambda\beta.\ (\lambda\gamma.\ a\ \times b\ \times\ y)))5$	
$(\lambda\beta.\ (\lambda\gamma.\ 5 \times b\ \times\ y))$	$\beta \rightarrow (\gamma \rightarrow x)$
$(\lambda\beta.\ (\lambda\gamma.\ 5 \times b\ \times\ y))7$	
$(\lambda\gamma.\ 5 \times 7\ \times\ y)$	$\gamma \rightarrow x$
$(\lambda\gamma.\ 5 \times 7\ \times\ y)3$	
$5 \times 7\ \times\ 3$	
105	x

Pela vantagem do currying ser em sua maioria teórica, muitas vezes novos programadores não são adeptos a isso, porém com a utilização do currying as provas formais são fáceis, pois todas as funções são tratadas uniformemente (um argumento entra e um resultado sai, sendo assim, mais fácil a compreensão das funções.

Referências Bibliográficas

Barendregt, Henk; Dekkers, Wil; Statman, Richard. **Lambda Calculus with Types.** Disponível em: <https://books.google.com.br/books?id=2UVasvrhXl8C&printsec=frontcover&hl=pt-BR&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false>. Acessado em: 09 ago. 2022

Riffel, Felipe K; Andrade, Robson J; Nogueira, Rodrigo R. **Desenvolvimento de uma Aplicação Interativa para ensino de Interações entre Seres Vivos e Dinâmica de Populações.** Disponível em: <http://www.etic.ifc-camboriu.edu.br/2018/pdf/9.pdf>. Acessado em: 10 ago. 2022

SILVA, André R. **Provador Interativo de Teoremas com Tipos Dependentes.** Disponível em: <http://dsc.inf.furb.br/arquivos/tccs/monografias/2015_1_andre-ramaciotti-da-silva_monografia.pdf>. Acessado em: 09 ago. 2022