

# Lista 02

## Programação Funcional

Prof. Maycon Amaro

### Orientações

- Qualquer estratégia de avaliação é aceitável para os exercícios.
- Utilize o interpretador em <https://lambster.dev> como suporte. Ele utiliza uma variação do *call-by-value* como estratégia de avaliação.
- Se quiser feedback sobre suas soluções, envie para `maycon.amaro@ufop.edu.br`, iniciando o assunto com [BCC222]

### Exercícios

1. Escreva um termo em  $\lambda$ -cálculo que realize a multiplicação de dois *church numerals* e mostre a avaliação de  $2 \times 3$ . Utilize `plus` como abreviação para o termo que soma dois *church numerals*.
2. Escreva a redução do termo abaixo, que representa `if 3 == 0 then 1 else 2`. As quebras de linha separam algumas aplicações para caber na margem.

$$\begin{aligned} &(\lambda y.((\lambda b.\lambda m.\lambda n.b\ m\ n) \\ &\quad ((\lambda n.n\ (\lambda x.\lambda t.\lambda f.f)\ (\lambda t.\lambda f.t))y)\ (\lambda s.\lambda z.sz)\ (\lambda s.\lambda z.s\ (sz)))) \\ &(\lambda s.\lambda z.s\ (s\ (sz))) \end{aligned}$$

Observação: Não se frustre se não estiver dando certo. Substituições no  $\lambda$ -cálculo são um processo que exige muita atenção quando executado por um humano.

3. Escreva um termo que realize a conjunção (o *e/and* lógico) de dois *church booleans*.
4. Escreva um termo que verifique se uma lista é vazia. Use o interpretador para conferir sua solução.

5. (Desafiador) Pesquise sobre o combinador Y e tente escrever uma função de fatorial em  $\lambda$ -cálculo. Tenha em mente que o combinador *call by name* causará um *loop infinito* se usado em um cenário *call by value*.
6. (Dark Souls) Escreva um termo que verifique qual dentre dois *church numerals* é o maior. Dica: além de recursão, você precisará da função de predecessor, que não é trivial. Ela utiliza a representação de tuplas, que não foi dada em aula. Você terá que pesquisar bastante.