

# Introdução ao Cálculo Lambda

Juliana Kaizer Vizzotto

Universidade Federal de Santa Maria

Linguagens de Programação

# Roteiro

- ▶ Semântica Operacional
- ▶ Formalidades

# Semântica Operacional

- ▶ Na sua forma pura, o cálculo- $\lambda$  não possui constantes nem operadores pré-definidos - não possui números, nem operações aritméticas, nem condicionais, nem *arrays*, nem *loops*, I/O, etc.
- ▶ A única coisa que temos para “computar” é a aplicação de funções em argumentos (os quais podem ser funções também).
- ▶ Cada passo na computação consiste de reescrever uma aplicação cujo componente esquerdo é uma abstração, substituindo a variável ligada no corpo da abstração pelo componente do lado direito:

$$(\lambda x. t_{12}) t_2 \rightarrow [x \mapsto t_2] t_{12},$$

- ▶ Onde  $[x \mapsto t_2] t_{12}$  significa “o termo obtido pela **substituição** de todas as ocorrências livres de  $x$  em  $t_{12}$  por  $t_2$ .”

# Semântica Operacional (Beta-reduction)

- ▶ Por exemplo, o termo  $(\lambda x.x)y$  avalia para  $y$ .
- ▶ O termo  $(\lambda x.x(\lambda x.x))(u\ r)$  avalia para  $u\ r(\lambda x.x)$ .
- ▶ De acordo com Church, um termo da forma  $(\lambda x.t_{12})t_2$  é chamado **redex** (ou “expressão redutível”).
- ▶ A operação de reescrever um redex de acordo com a regra acima é chamada **redução-beta** (**beta-reduction**).

# Estratégias de Avaliação

- ▶ Existem várias estratégias de avaliação para o cálculo- $\lambda$ .
- ▶ Cada estratégia define qual redex ou redexes em um termo podem ser reduzidos no próximo passo de avaliação:
  - ▶ *Full beta-reduction*: qualquer redex pode ser reduzido em qualquer momento. (Relação de avaliação não é uma função, pois um termo pode avaliar para mais de um termo em um passo)
  - ▶ *Normal order*: redex mais à esquerda e mais externo é sempre reduzido primeiro.
  - ▶ *Call-by-name*: não reduzir termos dentro de abstrações. (Haskell usa uma versão otimizada conhecida como call-by-need (não re-avalia termos).
  - ▶ *Call-by-value*: (maioria das linguagens) reduz somente os redexes mais externos e um redex é somente reduzido quando o seu lado direito já foi reduzido a um valor.

# Substituição: definição indutiva sobre o argumento $t$

- Primeira tentativa:

$$\begin{aligned}[x \mapsto s]x &= s \\ [x \mapsto s]y &= y \text{ se } x \neq y \\ [x \mapsto s](\lambda y. t_1) &= \lambda y. [x \mapsto s]t_1 \\ [x \mapsto s](t_1 t_2) &= ([x \mapsto s]t_1)([x \mapsto s]t_2)\end{aligned}$$

- Esta definição funciona para a maioria dos casos, por exemplo

$$[x \mapsto (\lambda z. zw)](\lambda y. x) = \lambda y. \lambda z. zw$$