Uma Implementação do Cálculo Lambda não Tipado em Elixir

Christian S. Lima[1], **Adolfo Neto**[1]

[1]Universidade Tecnológica Federal do Paraná (UTFPR), Curitiba, Brasil

christiansantoslima21@gmail.com, adolfo@utfpr.edu.br

Abstract.

Resumo. *Nesse artigo vamos fazer uma revisão bibliográfica sobre o cálculo lambda não tipado e apresentar uma implementação em Elixir.*

1. Introdução

2. Linguagem

Definição 1. O alfabeto do cálculo lambda é dado pelos seguintes símbolos:

- um conjunto de variáveis:

$$Var = \{x_i : i \in \mathbb{N}\};$$

- um abstrator: λ;
- três delimitadores: "(", ".", ")".

Definição 2. Os λ-termos são definidos de forma indutiva pelas regras:

1. todas as variáveis são λ-termos;
2. se M e N são λ-termos, então (MN) é um λ-termo (chamado de aplicação);
3. Se M é um λ-termo e x uma variável, então $(\lambda x.M)$ é um λ-termo (chamado abstração).

Definição 3. Definimos recursivamente o conjunto das variáveis que ocorrem livres em um λ-termo M pelas regras:

1. $\mathrm{FV}[x] = \{x\}$;
2. $\mathrm{FV}[NP] = \mathrm{FV}[N] \cup \mathrm{FV}[P]$;
3. $\mathrm{FV}[\lambda x.N] = \mathrm{FV}[N] - \{x\}$.

Definição 4. Definimos recursivamente a substituição de todas as ocorrências livres de x por N pelas regras:

1. $x[x := N] = N$;
2. $y[x := N] = y$, se $x \neq y$;
3. $(PQ)[x := N] = P[x := N]Q[x := N]$;

4. $(\lambda x.P)[x := N] = \lambda x.P$;
5. $(\lambda y.P)[x := N] = \lambda y.P$ se $x \notin \mathrm{FV}[P]$;
6. $(\lambda y.P)[x := N] = \lambda y.P[x := N]$ se $x \in \mathrm{FV}[P]$ e $y \notin \mathrm{FV}[N]$;
7. $(\lambda y.P)[x := N] = \lambda z.P[y := z][x := N]$ se $x \in \mathrm{FV}[P]$ e $y \in \mathrm{FV}[N]$.

Definição 5. Seja um termo P e que contém uma abstração $\lambda x.M$ como subfórmula e seja $y \notin \mathrm{FV}[M]$. Uma α-conversão de P é um termo Q obtido a partir de P substituindo uma ou mais ocorrências da subfórmula $\lambda x.M$ por $\lambda y.M[x := y]$.

3. Referências

Referências

Barendregt, H. P. (1984). *The Lambda Calculus: Its Syntax and Semantics*. Sole distributors for the U.S.A. and Canada, Elsevier Science Pub. Co., New York, N.Y.

Hindley, J. R. (1997). *Basic simple type theory*. Number 42. Cambridge University Press.

Hindley, J. R. and Seldin, J. P. (2008). *Lambda-calculus and combinators: an introduction*. Cambridge University Press.

Sørensen, M. H. and Urzyczyn, P. (2006). *Lectures on the Curry-Howard isomorphism*. Elsevier.