INE5416 - Paradigmas da Programação (2015/2)

Relatório 4: Cálculo Lambda - Parte 1

Caique Rodrigues Marques 13204303

O cálculo lambda é um formalismo algébrico para representar a abstração de funções e argumentos na lógica matemática descrito por Alonzo Church com base na lógica combinatória de Schönfinkel e na teoria da recursividade de Stephen Kleene. Este relatório contém as informações coletadas através de estudos sugeridos, dentre eles, o artigo original de Church: *An unsolvable problem of elementary number theory.*

Definições

O cálculo lambda foi desenvolvido por Alonzo Church como solução para o problema da computabilidade proposto por David Hilbert em 1928 que, assim como o artigo de Alan Turing, define as bases para a nova área da matemática, a Ciência da Computação. A solução de Church foi formulada para dar bases formais ao conceito de computação efetiva. O cálculo lambda é formado por uma expressão definida recursivamente.

Uma expressão lambda é composta por:

- Variáveis x_1, x_2, ..., x_n, ...
- Os símbolos de abstração lambda ("λ") e ponto (".")
- Parênteses ()

O conjunto de expressões lambda, A, pode ser definida indutivamente:

1. Se x é uma variável, então x∈A
2. Se x é uma variável e M∈A, então (λx.M)∈A
3. Se M, N∈ A, então (M, N)∈A

Instâncias da segunda regra são chamadas de abstrações e instâncias da terceira regra são chamadas de aplicações. Segue alguns exemplos:

- $\lambda x.x^2 + xy + z\ (3) = 9 + 3y + z$
- $\lambda x.x$
- $\lambda t.t + 2(\lambda y.y^2 + 2y + 1(3)) = \lambda t.t + 2\ (9 + 6 + 1) = \lambda t.t + 2\ (16) = 18$
- $\lambda y.\sqrt[2]{y}\ (t) = \lambda t.\sqrt[2]{t}$

Uma **abstração** é definida de forma que, usando o primeiro exemplo, λx corresponde ao lambda-termo e $x^2 + xy + z$ corresponde à expressão. O (3) corresponde à **aplicação** na expressão lambda, portanto, a aplicação do número três no lambda-termo, tal qual uma função matemática, onde dada uma função $f(x)$ qualquer, um valor qualquer presente no domínio de $f(x)$ pode ser aplicado em x. O cálculo lambda é a base das linguagens de programação com o paradigma funcional. A seguir, um exemplo em Python, que é uma linguagem multiparadigma. Nota-se que as funções anônimas do Python segue o mesma sintaxe e semântica provinda das definições do cálculo lambda.

```
def sqrt(n):
    return lambda n: n*n
```

A implementação acima, é o mesmo que $\lambda n.n^2\ (n)$, onde "n" corresponde a um número qualquer. Nota-se que o valor de λn corresponde à **variável** considerada na questão, enquanto (n) corresponde ao **argumento** que será usado na função, mesmo nome usado em linguagens ao definir a assinatura de uma função (assinatura corresponde a "def sqrt(n)", onde (n) é o argumento).

Reduções

Expressões em cálculo lambda também podem sofrer mundaças de comportamento, isto é chamado de redução e há três tipos:

- $\alpha-$conversão, onde se altera o limite das variáveis. Geralmente pode ser usada para alterar o nome da variável, embora as regras precisas não sejam triviais.
  Exemplo: $\lambda x.x\ (y)$

- $\beta-$redução, onde se aplica funções nos argumentos de uma expressão lambda.
  Exemplo: $\lambda y.y + 2\ (3x + 2y + 7z)$

- $\eta-$conversão, onde captura a noção de extensionalidade. Isto siginifica que, dada duas funções, elas só serão iguais se, e somente se, elas dão o mesmo resultado para todos os argumentos definidos.
  Exemplo: $\lambda x.(fx)$