Alfa-Equivalência

Francisco Handrick Tomaz da Costa, Luís Henrique Vieira Amaral, Diego Marcílio

Linguagens Declarativas, PPGI
Universidade de Brasília, Brasília-DF, Brasil

Ao fazermos a análise dessas 2 expressões, **λx . + (*2 x) y e λa . + (*2 a) b**, verifica-se que elas apresentam comportamento semelhantes, e que a única diferença entre elas se verifica justamente no nome que foi atribuído a seus parâmetros. Portanto, As duas expressões representam claramente a mesma função ou λ-expressão. Esse processo de renomear variáveis ligadas é conhecida como conversão alfa ou alfa renomeação. A conversão alfa nos permite trocar os nomes dos parâmetros formais de qualquer λ-abstração. Termos que diferem apenas por alfa-conversão são chamados alfa equivalente. Exemplo de alfa equivalência é apresentado pelo exemplo abaixo:

Exemplo 1.

$\lambda x . + (*2\ x)\ y \overset{\alpha}{\leftrightarrow} \lambda a . + (*2\ a)\ b$

Entretanto, a alfa conversão não é tão trivial, como aparentemente parece à primeira vista. Nem sempre trata-se de uma simples substituição de variáveis [1], como é possível observar pelo exemplo a seguir, onde é realizada a substituição do x pelo y.

λx . + x y transforma-se em **λy . + y y** , o que claramente não é a mesma λ-expressão, e portanto não são alfa equivalentes.

No cálculo lambda, a utilização de conversões alfas é de grande importância, uma vez que através delas, se torna possível resolver conflitos de nomes em λ-abstrações. Vejamos o que ocorre nesse outro exemplo:

Exemplo 2.

Definição de **z** = **λx . λy.(x y)**

Considere agora: **((z y) b)**, o que equivale a expressão: **((λx . λy.(x y) y) b)**

O que se observa é que o y, é usado tanto como variável ligada, como também como variável livre. Como variável ligada, ele será substituído através de β reduções, porém como variável livre, ele permanecerá o mesmo. A β redução é apresentada a seguir:

β redução.

((λx . λy.(x y) y) b) → λy.(y y) b → b b

O y foi substituído no escopo da variável ligada y, o que acarretou a criação de uma nova variável ligada, o que não era um resultado esperado. Para evitar esse tipo de comportamento, é possível se utilizar alfa conversão [2].

A expressão **((λx . λy.(x y) y) b)** é alfa equivalente a **((λx . λw.(x w) y) b),** onde foi substituído o y por w.

Aplicando a β redução temos:

β **redução.**

((λx . λw.(x w) y) b) → λw.(y w) b) → y b

Assim, uma alfa conversão poderá ser utilizado no cálculo lambida para remover os conflitos de nomes. Esse problema ocorre sempre quando uma β redução substitui uma expressão, com uma variável livre no escopo de uma variável ligada, com o mesmo nome da variável livre [2].

Referências:

1. B. C. Pierce, Types and programming languages. MIT press, 2002.
2. Robert Harper, Practical Foundations of Programming Languages. Carnegie Mellon University, 2006