

Trabalho de Linguagens declarativas

Gustavo van Erven

A) Alfa equivalência

As variáveis em uma expressão lambda são associadas ao corpo da função através do nome. Por exemplo, em:

$\lambda \text{func. } \lambda \text{arg.}(\text{func arg})$

A variável func no corpo da aplicação corresponde a variável λfunc . A variável arg é livre com relação à primeira abstração e ligada como parâmetro de λarg . Dessa forma, a aplicação dessa função a dois argumentos será reduzida a:

$(\lambda \text{func. } \lambda \text{arg.}(\text{func arg}) \text{ <argumento1> } \text{ <argumento2>})$
 $(\lambda \text{arg.}(\text{ <argumento1> arg}) \text{ <argumento2>})$
 $(\text{ <argumento1> } \text{ <argumento2>})$

Onde o argumento 1 foi ligado a func e o argumento 2 ao nome arg . Porém, considerando que argumento 1 tenha o mesmo nome da variável livre arg da primeira abstração:

$(\lambda \text{func. } \lambda \text{arg.}(\text{func arg}) \text{ arg}) \text{ <argumento2>}$
 $(\lambda \text{arg.}(\text{arg arg}) \text{ <argumento2>})$
 $(\text{ <argumento2> } \text{ <argumento2>})$

Se a redução beta for realizada simplesmente substituindo os nomes das variáveis ligadas, argumentos com o mesmo nome de variáveis livres em uma abstração podem gerar resultados incorretos no final da redução beta.

Dessa forma, sendo $\lambda \text{ <name> . body}$ uma abstração, toda a referência de <name> deve ser substituído no corpo body por uma referência <name2> que não seja igual ao nome do argumento durante a avaliação (variável livre).

B) Arquivos LambdaLibrary e Compiler

C) Arquivo Teste