Programming Languages Recitation
Lambda

Arpit Jain

Computer Science Department
Courant Institute of Mathematical Sciences
New York University

arpit.jain@cs.nyu.edu

October 9, 2014

Overview

Lambda Free Variables

- $(\lambda x.\lambda y.((\lambda z.x)(\lambda y.z)))$
- $(\lambda z.zz)z$
- $(\lambda x.\lambda y.\lambda z.(zy(\lambda w.x)))$
- $(\lambda x.w(\lambda w.(y(\lambda z.(f(\lambda f.f))))))$

Lambda Free Variables

- $(\lambda x.\lambda y.((\lambda z.x)(\lambda y.z)))$
- $(\lambda x.\lambda y.((\lambda z.{\color{red}x})(\lambda y.{\color{green}z})))$
- $(\lambda z.zz)z$

- $(\lambda x.\lambda y.\lambda z.(zy(\lambda w.x)))$

- $(\lambda x.w(\lambda w.(y(\lambda z.(f(\lambda f.f))))))$

Lambda Free Variables

- $(\lambda x.\lambda y.((\lambda z.x)(\lambda y.z)))$
- $(\lambda x.\lambda y.((\lambda z.x)(\lambda y.z)))$
- $(\lambda z.zz)z$
- $(\lambda z.zz)z$
- $(\lambda x.\lambda y.\lambda z.(zy(\lambda w.x)))$

- $(\lambda x.w(\lambda w.(y(\lambda z.(f(\lambda f.f))))))$

Lambda Free Variables

- $(\lambda x.\lambda y.((\lambda z.x)(\lambda y.z)))$
- $(\lambda x.\lambda y.((\lambda z.x)(\lambda y.z)))$
- $(\lambda z.zz)z$
- $(\lambda z.zz)z$
- $(\lambda x.\lambda y.\lambda z.(zy(\lambda w.x)))$
- $(\lambda x.\lambda y.\lambda z.(zy(\lambda w.x)))$
- $(\lambda x.w(\lambda w.(y(\lambda z.(f(\lambda f.f))))))$

Lambda Free Variables

- $(\lambda x.\lambda y.((\lambda z.x)(\lambda y.z)))$
- $(\lambda x.\lambda y.((\lambda z.x)(\lambda y.z)))$
- $(\lambda z.zz)z$
- $(\lambda z.zz)z$
- $(\lambda x.\lambda y.\lambda z.(zy(\lambda w.x)))$
- $(\lambda x.\lambda y.\lambda z.(zy(\lambda w.x)))$
- $(\lambda x.w(\lambda w.(y(\lambda z.(f(\lambda f.f))))))$
- $(\lambda x.w(\lambda w.(y(\lambda z.(f(\lambda f.f))))))$

Lambda Reductions

- PLUS [0] [2]
  $\Rightarrow ((\lambda m.\lambda n.\lambda fx.mf(nfx)[0])[2]))$
  $\Rightarrow (\lambda n).\lambda fx.[0]f(nfx)[2])$
  $\Rightarrow \lambda fx.[0]f([2]fx)$
  Using $[0] \Rightarrow (\lambda gy.y)$
  $\Rightarrow \lambda fx.\lambda gy.yf([2]fx)$
  $\Rightarrow \lambda fx.\lambda y.y([2]fx)$
  $\Rightarrow \lambda fx.[2]fx$
  Using $[2] \Rightarrow (\lambda zp.z(zp))$
  $\Rightarrow \lambda fx.\lambda zp.z(zp)fx$
  $\Rightarrow \lambda fx.\lambda p.f(fp)x$
  $\Rightarrow \lambda fx.f(fx) \equiv [2]$

- SUCC [2]
  $\Rightarrow \lambda nfx.f(nfx)[2]$
  $\Rightarrow \lambda fx.f([2]fx)$
  Using [2] $\Rightarrow (\lambda zp.z(zp))$
  $\Rightarrow \lambda fx.f(\lambda zp.z(zp))fx$
  $\Rightarrow \lambda fx.f(\lambda p.f(fp))x$
  $\Rightarrow \lambda fx.f(f(fx)) \equiv [3]$
- IF TRUE 1 0
  $\Rightarrow \lambda cte.(cte)TRUE10$
  $\Rightarrow \lambda t.\lambda e(\lambda a.\lambda b.ate)10$
  $\Rightarrow \lambda e(\lambda a.\lambda b.a1e)0$
  $\Rightarrow (\lambda a.\lambda b.a)10$
  $\Rightarrow (\lambda b.1)0$
  $\Rightarrow 1$

- OR FALSE FALSE

  $\Rightarrow \lambda mnab.ma(nab) FALSE$ FALSE

  $\Rightarrow \lambda nab.$ FALSE $a(nab) FALSE$

  $\Rightarrow \lambda ab.$ FALSE $a($ FALSE $ab)$

  Using FALSE $\Rightarrow \lambda ab.b$

  $\Rightarrow \lambda ab.(\lambda ab.b)a(\lambda ab.b)ab$

  $\Rightarrow \lambda ab.(\lambda b.b)(\lambda b.b)b$

  $\Rightarrow \lambda ab.(\lambda b.b)b$

  $\Rightarrow \lambda a.\lambda b.b \equiv$ FALSE

$\alpha - \beta$ Reductions

- $(\lambda x.wx)(\lambda x.wx)$

$\alpha - \beta$ Reductions

- $(\lambda x.wx)(\lambda x.wx)$

- Since there is no free variable in the second expression which is bound in the first expression so there is no need for α conversion. Here w in the second expression w is the free variable which is not bound in the first expression.

  Performing β Reduction

  $\Rightarrow w(\lambda x.wx)$

  When a argument will be given to this expression it will evaluated further.

- $(\lambda xy.xy)(\lambda x.y)$

- $(\lambda xy.xy)(\lambda x.y)$

- Here β Reduction cannot be performed without α conversion since the free variable y is bound in the first expression.

  Performing β Reduction (Correct way)

  First α Conversion of first expression

  $(\lambda xy.xy)\overrightarrow{\alpha}(\lambda xw.xw))$

  Now Using this along with second expression for β Reduction
  $\Rightarrow (\lambda xw.xw))(\lambda x.y)$

  $\Rightarrow \lambda w.\lambda x.yw$

  $\Rightarrow \lambda w.y$

- $(\lambda ab.cd)(\lambda cd.ab)$

- $(\lambda ab.cd)(\lambda cd.ab)$

- Here α Conversion is required since the free variables in the second expression are bound in the first expression.