Lambda Calculus

Dr. Mattox Beckman

University of Illinois at Urbana-Champaign
Department of Computer Science

Objectives

You should be able to ...

The purposes of this lecture is to introduce lambda calculus and explain the role it has in programming languages.

- Explain the three constructs of λ-calculus.
- Given a syntax tree diagram, write down the equivalent λ-calculus term.
- Perform a beta-reduction.

The λ-Calculus

- Contains three kinds of things:

The λ-Calculus

- Contains three kinds of things:
  1. Variables: x y_3 z' – usually we assume they are all one letter long.

The λ-Calculus

- Contains three kinds of things:
  1. Variables: $x\ y_3\ z'$ – usually we assume they are all one letter long.
  2. Function application:

$$
\begin{array}{l}
fy \\
abc \\
x(fy)(fg)
\end{array}
$$

The λ-Calculus

- Contains three kinds of things:
  1. Variables: x y_3 z' – usually we assume they are all one letter long.
  2. Function application:
     $$fy$$
     $$abc$$
     $$x(fy)(fg)$$
  3. Functions (Also called *abstractions*.)
     $$\lambda x.x$$
     $$\lambda ab.fab$$
     $$\lambda xy.g(\lambda z.zf)yx$$

The λ-Calculus

- Contains three kinds of things:
  1. Variables: x y_3 z' – usually we assume they are all one letter long.
  2. Function application:
  $$fy$$
  $$abc$$
  $$x(fy)(fg)$$
  3. Functions (Also called *abstractions*.)
  $$\lambda x.x$$
  $$\lambda ab.fab$$
  $$\lambda xy.g(\lambda z.zf)yx$$
- Used extensively in research. The "little white mouse" of computer science.
- We can implement this trivially in Haskell.
  $\lambda x.x$ = `\x -> x`.

Examples

$\lambda x.x$ “The identity”

$\lambda x.xx$ “Delta”

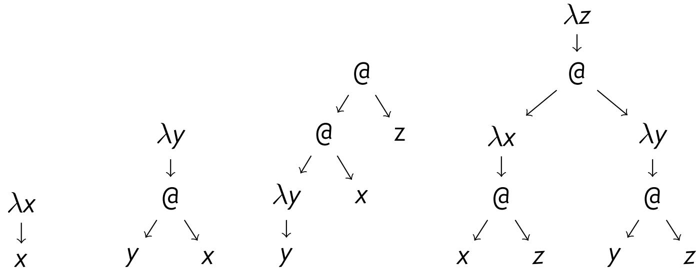
$\lambda ab.fabxy$

$(\lambda ab.fab)xy$

$(\lambda a.\lambda b.fab)xy$

$(\lambda fx.xf)(\lambda g.gx)(\lambda f.f)zy$

Syntax Trees

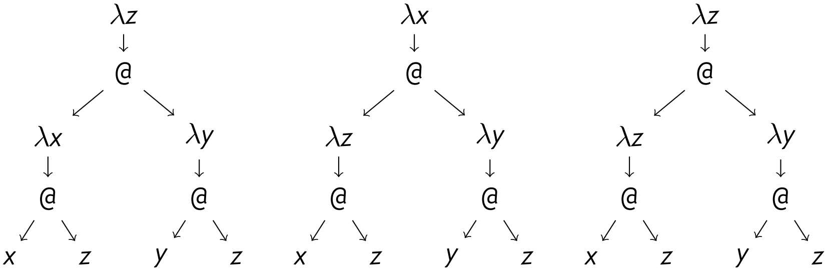


Example 1	Example 2	Example 3	Example 4
$\lambda x.x$	$\lambda y.yx$	$(\lambda y.y)xz$	$\lambda z.(\lambda x.xz)(\lambda y.yz)$

Bound and Free

- The λ creates a *binding*.
- An occurance of the the variable inside the function body is said to be *bound*.
- Bound variables occur "under the λ" that binds them.

Examples: Where are the free variables? To which lambdas are bound variables bound?



$\lambda z.(\lambda x.xz)(\lambda y.yz)$ $\quad$ $\lambda x.(\lambda z.xz)(\lambda y.yz)$ $\quad$ $\lambda z.(\lambda z.xz)(\lambda y.yz)$

Function Application

$$(\lambda x.M)N \quad \mapsto \quad [N/x]M$$

$$[N/x]\, y = y$$
$$[N/x]\, x = N$$
$$[N/x]\, (M\ P) = ([N/x]M\ [N/x]P)$$
$$[N/x]\, (\lambda y.M) = \lambda y.[N/x]M$$
$$[N/x]\, (\lambda x.M) = \lambda x.M$$