COM S-342

Recitation 10/22/18 – 10/24/18

Today

- Q & A lambda calculus
- Exercises on lambda calculus

Help from: http://www-verimag.imag.fr/~iosif/LogicAutomata07/lambda-calculus-slides.pdf

Lambda Calculus

- Formal mathematical system
- Simplest programming language
- Invented in 1936 by Alonzo Church (1903-1995)

Lambda Calculus

- Syntax implemented in Scheme: $(\lambda (x)\ e)$

e	$\rightarrow$	x	Variable
	$\mid$	$(\lambda\ (x)\ e)$	a lambda expression
	$\mid$	$(e\ e)$	Application

- Only functions
- Arguments are functions
- Returned values are functions

Beta Reduction

- $((_x\ \lambda\ (x)\ e_1\ )_x\ e_2)$: Evaluate the expression e_1 by replacing every ("free") occurrences of x in e_1 by e_2. I.e., $e_1[x \mapsto e_2]$ (β-reduction)

$$(\ (_x\ \lambda\ (x)\ (_y\ \lambda\ (y)\ (+\ x\ y)\ )_y\ )_x\ 1)$$

$$(_y\ \lambda\ (y)\ (+\ x\ y)\ )_y[x \mapsto 1]$$

$$(_y\ \lambda\ (y)\ (+\ 1\ y)\ )_y$$

Beta Reduction

$$\text{(a)} \quad ((\lambda(z)z)((\lambda(y)(y\ y))((\lambda(x)x)a)))$$

$$\text{(b)} \quad (((\lambda(x)(\lambda(y)(y\ y)))(\lambda(a)a))b)$$

$$\text{(c)} \quad (((\lambda(x)(\lambda(y)(x\ y)))w)z)$$

Beta Reduction

a)

$$
\begin{aligned}
& ((\lambda(z)z)((\lambda(y)(y\,y))((\lambda(x)x)a))) && (1)\\
= \quad & ((\lambda(z)z)((\lambda(y)(y\,y))a)) && (2)\\
= \quad & ((\lambda(z)z)(a\,a)) && (3)\\
= \quad & (a\,a) && (4)
\end{aligned}
$$

Beta Reduction

b)

$$
\begin{aligned}
& (((\lambda(x)(\lambda(y)(y\,y)))(\lambda(a)a))b) && (7)\\
= \quad & ((\lambda(y)(y\,y))b) && (8)\\
= \quad & (b\,b) && (9)
\end{aligned}
$$

Beta Reduction

c)

$$
\begin{aligned}
& ((\lambda(x)(\lambda(y)(x\ y))w)z) && (10) \\
= \quad & ((\lambda(y)(w\ y))z) && (11) \\
= \quad & (w\ z) && (12)
\end{aligned}
$$

Natural Numbers

- A programming language should be capable of doing arithmetic
- Numbers can be represented in lambda calculus starting from zero,
- and writing “suc(zero)” to represent 1, “suc(suc(zero))” to represent
- Encoded by the number of applications of some function on some entity

Natural Numbers

$$\begin{array}{rl}
\text{zero} & (_f\ \lambda\,(f)\ (_x\ \lambda\,(x)\ x\ )_x\ )_f \\
\text{one} & (_f\ \lambda\,(f)\ (_x\ \lambda\,(x)\ (f\ x)\ )_x\ )_f \\
\text{two} & (_f\ \lambda\,(f)\ (_x\ \lambda\,(x)\ (f\ (f\ x))\ )_x\ )_f \\
n & (_f\ \lambda\,(f)\ (_x\ \lambda\,(x)\ (f\ \ldots(f\ x)\ldots))\ )_x\ )_f
\end{array}$$

Example

- ((two g) z): two applications of g on z

$$(((_f\ \lambda\ (f)\ (_x\ \lambda\ (x)\ (f\ (f\ x))\ )_x\ )_f\ g)\ z) = (g\ (g\ z))$$