Lambda Calculus

H. Geuvers

Institute for Computing and Information Sciences
Radboud University Nijmegen

Version: fall 2015

Intended meaning

The meaning of

$$\lambda x.x^2$$

is the function

$$x \longmapsto x^2$$

that assigns to x the value x^2 (x times x)
So according to this intended meaning we have

$$(\lambda x.x^2)(6) = 6^2 = 36.$$

The parentheses around the 6 are usually not written:

$$(\lambda x.x^2)6 = 36$$

Principal axiom is the β-eqality:

$$(\lambda x.M)N =_\beta M[x := N]$$

Alphabet: $\Sigma = \{x, ', (,), ., \lambda, =\}$
Language: the set of lambda terms, Λ:

$$\begin{array}{rcl} \texttt{variable} & := & \texttt{x} \mid \texttt{variable}' \\ \texttt{term} & := & \texttt{variable} \mid (\texttt{term}\,\texttt{term}) \mid (\lambda\,\texttt{variable}\,.\,\texttt{term}) \\ \texttt{formula} & := & \texttt{term} = \texttt{term} \end{array}$$

Theory (we often write just $=$ for $=_\beta$)

Axioms	$(\lambda x.\, M)N$	$=_\beta$	$M[x := N]$
	M	$=_\beta$	M
Rules	$M =_\beta N$	$\Rightarrow$	$N =_\beta M$
	$M =_\beta N, N =_\beta L$	$\Rightarrow$	$M =_\beta L$
	$M =_\beta N$	$\Rightarrow$	$ML =_\beta NL$
	$M =_\beta N$	$\Rightarrow$	$LM =_\beta LN$
	$M =_\beta N$	$\Rightarrow$	$\lambda x.\, M =_\beta \lambda x.\, N$

M	$M[x := N]$
x	N
y	y
PQ	$(P[x := N])(Q[x := N])$
$\lambda x.\, P$	$\lambda x.\, P$
$\lambda y.\, P$	$\lambda y.\,(P[x := N])$

where $y \not\equiv x$

Application associates to the left

$$P\,Q_1 \ldots Q_n \equiv (\ldots((P\,Q_1)\,Q_2)\ldots Q_n).$$

Abstraction associates to the right

$$\lambda x_1 \ldots x_n.M \equiv (\lambda x_1.(\lambda x_2.(\ldots(\lambda x_n.M)\ldots)))).$$

Outer parentheses are often omitted. For example

$$(\lambda x.x)y \equiv ((\lambda x.x)y)$$

Bound and free variables

$\lambda x.x$ and $\lambda y.y$ acting on M both give M

Renaming bound variables

- In the term $\lambda x.M$, the 'λx' binds the x in M.
- Variables can occur free or bound.
- We don't want to distinguish between terms that only differ in their bound variables
- We write $M \equiv_\alpha N$ (or just $M \equiv N$) if N arises from M by renaming bound variables

Examples

- $\lambda x.x \equiv_\alpha \lambda y.y$
- $\lambda x\, y.x \equiv_\alpha \lambda y\, x.y$
- $\lambda x.(\lambda x.x)\, x \equiv_\alpha \lambda y.(\lambda x.x)\, y$
- $(\lambda x.(\lambda y.x\, y))\, x \equiv_\alpha (\lambda z.(\lambda y.z\, y))\, x$

- $P[x := N]$ is only allowed if no free variable in N becomes bound after substitution.
- Otherwise: rename bound variables first.

$$(\lambda x.\lambda y.x\,y)\,(y\,y) =_\beta (\lambda y.x\,y)[x := y\,y]$$

$$(\ \equiv\ ??\ \lambda y.y\,y\,y\ \text{NO!!}\ ) \equiv (\lambda z.x\,z)[x := y\,y] \equiv \lambda z.y\,y\,z$$

$$\begin{array}{lll} \mathbf{K}yz & \equiv & (\lambda x.(\lambda y.\ x))\,y\,z \\ & =_\beta ?? & (\lambda y.\ y)\,z \\ & =_\beta & z?? \end{array}$$

better:

$$\begin{array}{lll} \mathbf{K}yz & \equiv & (\lambda x.(\lambda y'.\ x))\,y\,z \\ & =_\beta & (\lambda y'.\ y)\,z \\ & =_\beta & y \text{ as it should.} \end{array}$$

$$
\begin{array}{rcllrcl}
\mathbf{I} & \equiv & \lambda x.x & \Rightarrow & \mathbf{I} M & =_\beta & M \\
\mathbf{K} & \equiv & \lambda x\, y.x & \Rightarrow & \mathbf{K} M\, P & =_\beta & M \\
\mathbf{S} & \equiv & \lambda x\, y\, z.x\, z\, (y\, z) & \Rightarrow & \mathbf{S} M\, P\, Q & =_\beta & M\, Q\, (P\, Q) \\
\mathbf{D} & \equiv & \lambda x.x\, x & \Rightarrow & \mathbf{D} M & =_\beta & M\, M \\
\mathbf{B} & \equiv & \lambda x\, y\, z.x\, (y\, z) & \Rightarrow & \mathbf{B} M\, P\, Q & =_\beta & M\, (P\, Q)
\end{array}
$$

Reduction

The equations can be ordered into computation or reduction rules
There is a one-step reduction $\rightarrow$, more-step reduction $\twoheadrightarrow$ (0, 1 or more steps).

Axiom	$(\lambda x.M)\,N \rightarrow M[x := N]$		
Rules for $\rightarrow$	$M \rightarrow N$	$\Rightarrow$	$M\,Z \rightarrow N\,Z$
	$M \rightarrow N$	$\Rightarrow$	$M\,Z \rightarrow N\,Z$
	$M \rightarrow N$	$\Rightarrow$	$\lambda x.M \rightarrow \lambda x.N$
Rules for $\twoheadrightarrow$	$M \twoheadrightarrow M$		
	$M \rightarrow N$	$\Rightarrow$	$M \twoheadrightarrow N$
	$M \twoheadrightarrow N \wedge N \twoheadrightarrow L$	$\Rightarrow$	$M \twoheadrightarrow L$

Examples:

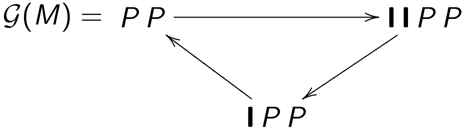
$$\begin{array}{rcl} \mathbf{I}\,x & \rightarrow & x. \\ \mathbf{I}\,\mathbf{I}\,x & \rightarrow & \mathbf{I}\,x \\ & \rightarrow & x. \\ \mathbf{I}\,\mathbf{I}\,x & \twoheadrightarrow & x. \end{array}$$

Reduction Graph

Given $M \in \Lambda$, the graph of M, $\mathcal{G}(M)$, is

$$\{N \mid M \twoheadrightarrow N\}$$

with $\rightarrow$ as the edges and the ‘reducts’ of M as the vertices
For example let $P \equiv \lambda x.\mathbf{I}\,\mathbf{I}\,x\,x$ and $M \equiv P\,P$. Then



Fixed point theorem

THEOREM. For all $F \in \Lambda$ there is an $M \in \Lambda$ such that

$$F\,M =_\beta M$$

PROOF. Defines $\mathbf{W} \equiv \lambda x.F\,(x\,x)$ and $M \equiv \mathbf{W}\,\mathbf{W}$. Then

$$\begin{array}{rcl} M & \equiv & \mathbf{W}\,\mathbf{W} \\ & \equiv & (\lambda x.F\,(x\,x))\,\mathbf{W} \\ & =_\beta & F\,(\mathbf{W}\,\mathbf{W}) \\ & \equiv & F\,M. \quad \Box \end{array}$$

COROLLARY. For any 'context' $C[\vec{x}, m]$ there exists a M such that

$$M\,\vec{P} =_\beta C[\vec{P}, M] \text{ for all terms } \vec{P}$$

PROOF. M can be taken the fixed point of $\lambda m\,\vec{x}.C[\vec{x}, m]$.
Then $M\,\vec{P} =_\beta (\lambda m\,\vec{x}.C[\vec{x}, m])\,M\,\vec{P} =_\beta C[\vec{P}, M]$. $\Box$

Using the Fixed Point Theorem

THEOREM. There is a Fixed Point Combinator $\mathbf{Y}$, that produces a fixed point for every term:

$$\mathbf{Y}\,F =_\beta F\,(\mathbf{Y}\,F) \text{ for all } F \in \Lambda.$$

PROOF. We have seen that, defining $\mathbf{W} \equiv \lambda x.F\,(x\,x)$, we get $M \equiv \mathbf{W}\,\mathbf{W}$ as a fixed point of F. So the following term is a fixed point combinator:

$$\mathbf{Y} := \lambda f.(\lambda x.f\,(x\,x))\,(\lambda x.f\,(x\,x)). \quad \square$$

Examples: We can construct terms $\mathbf{L}$, $\mathbf{O}$, $\mathbf{P}$ such that

$$\begin{array}{rcll} \mathbf{L} & =_\beta & \mathbf{L}\,\mathbf{L} & \text{take } \mathbf{L} \equiv \mathbf{Y}\,\mathbf{D}; \\ \mathbf{O}\,x & =_\beta & \mathbf{O} & \text{take } \mathbf{O} \equiv \mathbf{Y}\,\mathbf{K}; \\ \mathbf{P} & =_\beta & \mathbf{P}\,x. & \end{array}$$